RL-TR-93-117, Vol II (of two)
Final Technical Report
June 1993

# CADBIT II - COMPUTER-AIDED DESIGN FOR BUILT-IN TEST

Hughes Missile Systems Company

Michael Davis, Sonny Kwan, Tony Holzer

*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-93-117, Vol II (of two) has been reviewed and is approved for publication.

APPROVED:

EDWARD L. DEPALMA
Project Engineer

FOR THE COMMANDER: Anthony J Feduccia

ANTHONY J. FEDUCCIA
Acting Director, Reliability
Electromagnetics and Reliability Directorate

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
June 1993

3. REPORT TYPE AND DATES COVERED
Final - - -

4. TITLE AND SUBTITLE
CADBIT II - COMPUTER-AIDED DESIGN FOR BUILT-IN TEST

5. FUNDING NUMBERS
C - F30602-90-C-0116
PE - 62702F
PR - 2338
TA - 02
WU - 4U

6. AUTHOR(S)
Michael Davis, Sonny Kwan, Tony Holzer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Hughes Missile Systems Company
P.O. Box 85357
San Diego CA 92186

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (ERSD)
525 Brooks Road
Griffiss AFB NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-93-117, Vol II
(of two)

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Edward L. DePalma/ERSD/(315)330-2702.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The objective of CADBIT II was to develop a CAD workstation-based system with software to help designers incorporate Built-In Test (BIT) in their CAD-based circuit board designs. The CADBIT II system consists of a Unix-based workstation, electronic CAD software, and the contractor-developed CADBIT II software module. CADBIT II benefits printed circuit board designers through four major functions. The Tutorial function increases awareness and understanding of Built-In Test techniques. This function offers on-line access to an extensive database of information on current techniques used to implement built-in test in digital, analog, and hybrid (mixed digital and analog) circuit boards. The Selection function helps designers choose the best technique for their design. The selection log compares characteristics and attributes of each BIT technique in the CADBIT database to information about the user's specific "Circuit Under Test". The CAD Insertion function helps designers implement a selected BIT technique in a CAD schematic. This function provides a "Default Design: with a consistent set of components to implement the selected BIT technique. The Evaluation function analyzes the design impact of adding Built-In-Test circuitry based on total board area, weight, and power "penalties" for each BIT.

14. SUBJECT TERMS
Computer Aided Design, Built-In-Test, BIT, BILBO, Comparator, Software Design Tool, Mentor Graphics, Testability

15. NUMBER OF PAGES
290

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
U/L

# TABLE OF CONTENTS

**Section** **Page**

# LIST OF FIGURES

## LIST OF FIGURES, Continued

## LIST OF FIGURES, Continued

**Section** **Figure** **Page**

PSEUDORANDOM PATTERN GENERATOR WITH MULTIPLE INPUT SIGNATURE REGISTER (PRPG/MISR)

COMPARATOR

VOLTAGE SUMMING

REDUNDANCY

## LIST OF FIGURES, Continued

# LIST OF TABLES

## LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| A/D | Analog-to-Digital |
| ALS | Advanced Low-Power Schottky |
| ALU | Arithmetic and Logic Unit |
| ASIC | Application-Specific Integrated Circuit |
| BILBO | Built-In Logic Block Observer |
| BIST | Built-In Self-Test |
| BIT | Built-In Test |
| BTID | BIT Technique Insertion Diagram |
| CAD | Computer-Aided Design |
| CADBIT | Computer-Aided Design for Built-In Test |
| CDE | Component Determination Equation |
| CDRL | Contract Data Requirements List |
| ceil | ceiling (C function to round up to nearest integer) |
| CLIN | Contract Line Item Number |
| CMRR | Common Mode Rejection Ratio |
| CPU | Central Processing Unit |
| CUT | Circuit Under Test |
| D/A | Digital-to-Analog |
| DA | Differential Amplifier |
| DIP | Dual In-line Package |
| ECC | Error Correction Code |
| EDC | Error Detection and Correction |
| EDCC | Error Detection and Correction Code |
| EDCU | Error Detection and Correction Unit |
| EEPROM | Electrically-Erasable Programmable Read-Only Memory |
| ELIN | Exhibit Line Item Number |
| EPROM | Erasable Programmable Read-Only Memory |
| FF | Flip-Flop |
| FR | Final Report |
| GMR | Good Machine Response |
| gms | grams |
| HMSC | Hughes Missile Systems Company |
| IC | Integrated Circuit |

## LIST OF ACRONYMS AND ABBREVIATIONS, Continued

| | |
|---|---|
| I/F | Interface |
| LFSR | Linear Feedback Shift Register |
| LRM | Line Replaceable Module |
| LS | Low-Power Schottky |
| LSI | Large-Scale Integration |
| LSSD | Level-Sensitive Scan Design |
| MHz | Megahertz |
| MIL-STD | Military Standard |
| MISR | Multiple-Input Signature Register |
| MSB | Most Significant Bit |
| MSI | Medium-Scale Integration |
| MUX | Multiplexer |
| mW | milliwatt |
| ns | nanosecond |
| OBIVCB | On-Board Integration of VLSI Chip BIT |
| PCB | Printed Circuit Board |
| PISO | Parallel-In, Serial-Out |
| PROM | Programmable Read-Only Memory |
| PRPG | Pseudorandom Pattern Generation |
| RAM | Random Access Memory |
| RF | Radio Frequency |
| RL | Rome Laboratory |
| RMS | Root-Mean-Square |
| ROM | Read-Only Memory |
| SRAM | Static Random Access Memory |
| SSI | Small-Scale Integration |
| S/W | Software |
| TIM | Technical Interchange Meeting |
| TP | Test Pattern |
| upp | units per package |
| URD | User Requested Data |
| VHDL | VHSIC Hardware Description Language |
| VHSIC | Very High Speed Integrated Circuit |
| VLSI | Very Large Scale Integration |

## 1.0 INTRODUCTION

The Final Report was prepared as part of contract number F30602-90-C-0116, Computer-Aided Design for Built-In-Test II (CADBIT II) between Rome Laboratory (RL) and the Hughes Missile Systems Company (HMSC). This report has been prepared to satisfy the requirement for Contract Line Item Number (CLIN) 0002, Exhibit Line Item Number (ELIN) A007.

This is Volume II of the CADBIT II Final Report and contains the CADBIT II BIT Library Package. This package is included to satisfy the requirements of paragraph 4.1.1 of the CADBIT II Statement of Work (SOW). This package consists of a library of BIT techniques in a standard format which was encoded and stored as the database of CADBIT functions.

The purpose of the CADBIT II system is to provide an automated procedure to aid the electronic circuit designer in the selection of BIT techniques, the insertion of the associated BIT circuitry into the Printed Circuit Board (PCB) design, and to provide a post design evaluation of the penalties incurred by the addition of BIT circuitry into the PCB functional design.

# 2.0 DESCRIPTION OF CADBIT II BIT LIBRARY

## 2.1 BIT TECHNIQUES

A list of BIT techniques described in this volume are classified below as Digital, Analog, or Hybrid and briefly summarized.

DIGITAL techniques:

**ON BOARD ROM** —BIT test patterns and good machine responses are stored in on board ROM. As each test pattern is applied the output of the Circuit Under Test (CUT) is compared to the known good machine responses. A mismatch indicates a test failure.

**MICROPROCESSOR BIT** — By storing the BIT program in ROM, a microprocessor can test itself and the associated peripheral circuitry such as memory and Input/Output (I/O).

**MICRODIAGNOSTICS** — Microprogrammed processors have their program stored in external ROM. BIT instructions can be appended to the program resident in external ROM to perform test for both internal and external processor hardware. Currently, this technique is rarely used since bit slice design becomes less popular.

**ON BOARD INTEGRATION OF VLSI CHIPS BIT (OBIVCB)** — Many VLSI chips are designed with built in BIT circuitries such as scan set and BILBO. This technique shows how to control and integrate these internal BIT chips properly on the board level design.

**BUILT IN LOGIC BLOCK OBSERVER (BILBO)** — With minor modification of the existing registers in a design and two added control lines, this technique allows full access to the internal circuit of the CUT. By manipulating the two control lines, the CUT can be put under four different modes of operation. The four modes are scan, latch, reset and PRPG/MISR.

**ERROR DETECTION AND CORRECTION CODES** — This is a proven technique that can greatly improve memory system reliability through its ability to detect and correct memory errors. Currently this technique can be implemented using off the shelf IC.

SCAN — In practice, there are many different ways to do scan. In this technique, test vectors are stored in an EEPROM and a maintenance processor is used to control the data flow in and out the serial scan chain associated with the CUT.

**DIGITAL WRAPAROUND** — The normal data flow in a microprocessor system can be short circuited by using a wraparound gate across the input and output devices next to the microprocessor. In this case, the microprocessor serves as part of the CUT, test controller, test pattern generator and response comparator.

**PSEUDORANDOM PATTERN GENERATOR / MULTIPLE INPUT SIGNATURE REGISTER (PRPG/MISR)** — In this technique, a linear feedback shift register (LFSR) is used to generate pseudorandom test patterns applied to the CUT. The responses are captured by the MISR during the test. At the end of the test, the final state of the MISR called the signature is then compared to the predetermined proper signature to determine a pass or fail status of the CUT.

ANALOG techniques:

**COMPARATOR** — Analog test signals are fed to the CUT and its response is compared to a window comparator to determine whether the CUT is operating properly.

**VOLTAGE SUMMING** — Analog signal such as dc power supply voltages can be tested in this technique. The voltages are summed using an opamp summer circuit and the output is compared to a window comparator to determine whether the CUT is operating properly.

**REDUNDANCY** — The input stimulus is applied to both the CUT and the electrical equivalent of the CUT. Both outputs are fed to a differential amplifier and the difference is amplified. The amplified output is compared to a window comparator to determine pass/fail of the CUT.

HYBRID (mixed DIGITAL and ANALOG) techniques:

**ANALOG WRAPAROUND** — The normal data flow in a mixed signal microprocessor system can be short circuited by using a wraparound analog switch across the D/A and A/D converters. Again, similar to the digital wraparound BIT technique, the microprocessor serves as part of the CUT, test controller, test pattern generator and response comparator.

## 2.2 ELEMENTS OF BIT LIBRARY

The BIT Library contains data packages for all of the BIT techniques available with CADBIT II. The data is used by the CADBIT II software for tutorial presentation, BIT selection, insertion and evaluation. The following is a list of the data elements with a brief description of each:

**SHORT DESCRIPTION / DIAGRAMS** — This is an overview and brief description of the BIT technique with high-level block diagrams. The Level I Block Diagram illustrates the basic concept behind the BIT technique. The Level II Block diagram provides more functional detail and signal flow information and shows conceptually how to implement the BIT technique and set up correct control signals.

**FLOW CHART / DESCRIPTION** — The flow chart shows the proper sequence of the BIT when it is initiated in a typical design. It displays all the events among signal flow and component function from the start of the BIT to the end of it. The description provides a step-by-step text description of the BIT sequence shown in the Flow Chart

**ADVANTAGES** — This is a list of the advantages of using the BIT technique. It describes the benefits and design improvements in term of system reliability and testability.

**DISADVANTAGES** — Limitations and disadvantages of the BIT technique are explored in this section. User can gain insight and understand the pitfalls of the technique.

**ATTRIBUTES** — There are sixteen attributes listed in this section. The first four attributes are used to determine the suitable BIT techniques in the Suitability BIT Selection Process. The area, weight, power and timing penalties are used to calculate BIT technique penalty downstream. The other attributes are general information related to the BIT technique.

Concurrency
Technology
CUT Microprocessor Required?
CUT Internal Design Required?
Area Penalty
Weight Penalty
Power Penalty
Timing Penalty
Reliability Impact
Conceptual Complexity
Hardware / Software / Firmware
Design Cost
Memory Requirements
BIT Circuitry Self-Testable
Stand-alone (Self-Contained) BIT?
Notes

**DEFAULT DESIGN** — The Default Design includes a complete schematic built using Mentor Graphics Version 8 software, and a text description of the design.

**BIT TECHNIQUE INSERTION DIAGRAM (BTID)** — The BTID is a tool to assist users during the insertion process. It assists users in inserting BIT components and connecting them into the functional design. The diagram is supplemented with connection instructions to guide users in connecting BIT components to the target design. New users of CADBIT II will find this assistance valuable.

**VARIABLE DEFINITIONS** — CADBIT II needs this information about the Circuit Under Test so that it can generate the BIT Technique Penalty Report. The User Requested Data window is a list of questions merged from the Master Question List required by all the suitable BITs.

**COMPONENT DETERMINATION EQUATIONS (CDE)** — Equations to calculate the number of units (as represented by individual CAD symbols) and component parts (physical packages) required to implement the BIT technique using the Default Design. A single 'and' gate would be an example of a unit. The Unit Determination Equations allow the correct number of component symbols to be popped up in Design Architect during BIT insertion. Based on how many units of a device type are needed and how many units in a package, the Component Determination Equation calculates the number of packages required. A quad '2-input and' gate 54ALS08 has four units in it. The CDE provides the correct number of packages for area, power and weight calculations during BIT penalty ranking process.

**PENALTY EQUATIONS** — Penalties for board area, weight, typical power, test time, and throughput delay are defined. Test time is the amount of time required to perform the BIT. Throughput delay is the induced timing delay for signal flow caused by the additional BIT circuitry.

## 3.0 BIT TECHNIQUE DATA PACKAGES

This section contains data packages for the following BIT techniques, in the order shown below.

On-board ROM
Microprocessor
Microdiagnostics
On-board Integration of VLSI Chip BIT (OBIVCB)
Built-in Logic Block Observer (BILBO)
Error Detection and Correction Code (EDCC)
Scan Design
Digital Wraparound
Pseudorandom Pattern Generator with Multiple Input Signature Register (PRPG/MISR)
Comparator
Voltage Summing
Redundancy
Analog Wraparound

# ON-BOARD ROM

# BIT TECHNIQUE DATA PACKAGE

**CATEGORY: SHORT DESCRIPTION / DIAGRAMS**

On-Board Read Only Memory (ROM) Self-Test is a nonconcurrent Built-In-Test (BIT) technique implemented in hardware and firmware. The technique consists of applying test patterns that are stored in an on-board ROM to a Circuit Under Test (CUT), and then comparing the CUT's response to what is expected, resulting in a go - no/go output signal. Each test pattern can be individually and selectively determined, thereby maximizing the percentage of fault coverage while minimizing the number of test patterns.

Figures 1 and 2 provide Level I and II Block Diagrams for this BIT technique.

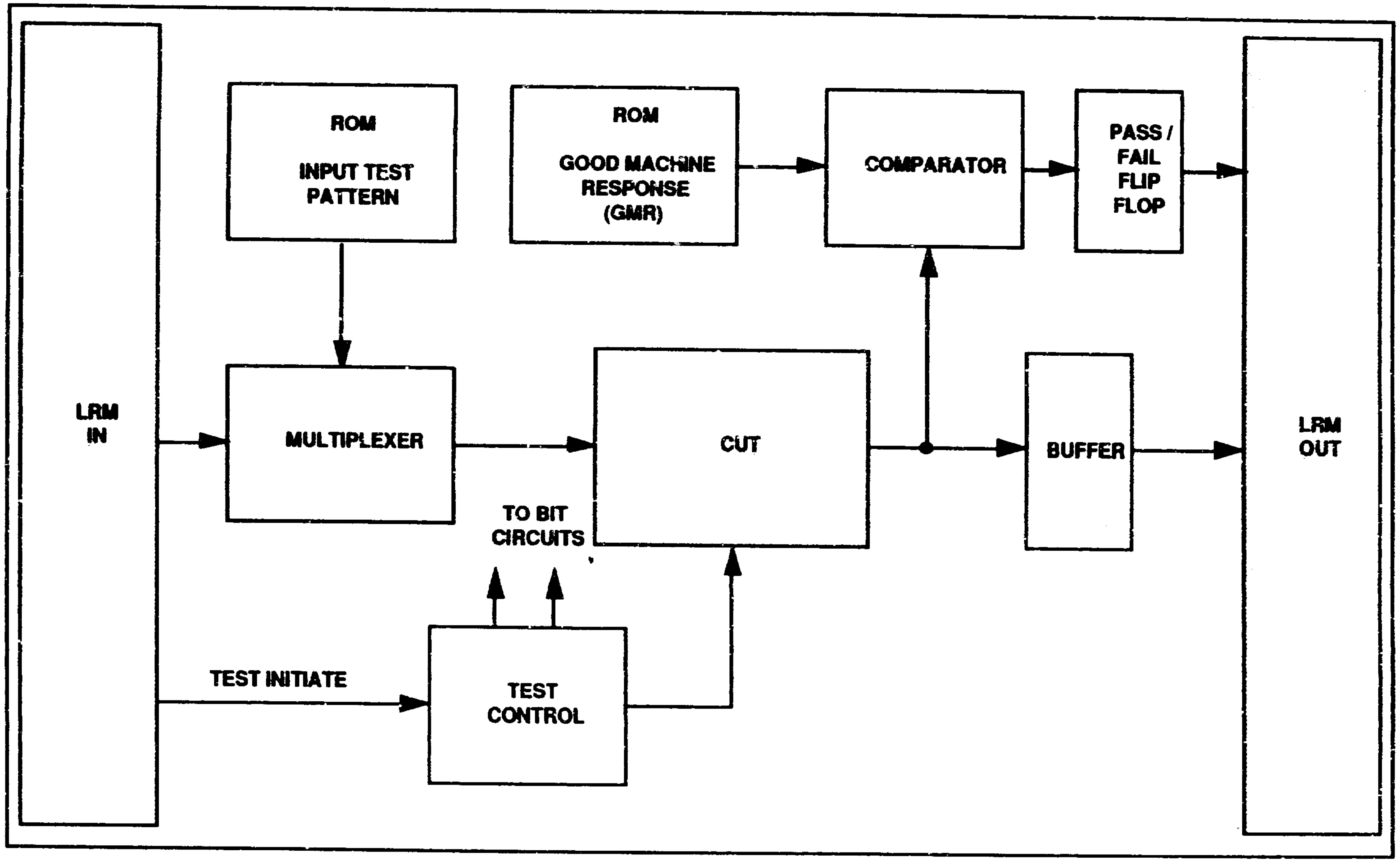
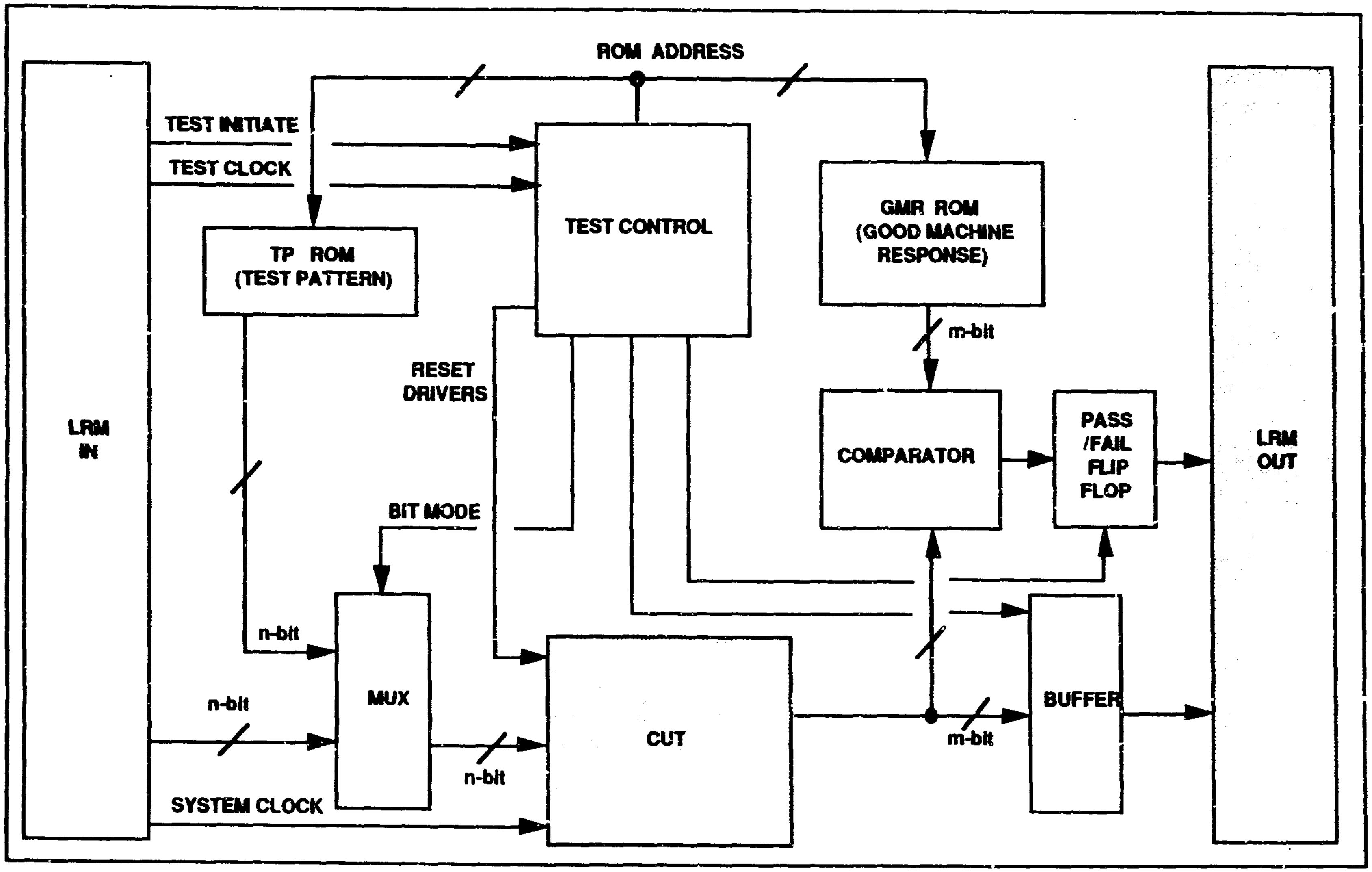


Figure 1 Level I Block Diagram for On-Board ROM



274_844_52

Figure 2 On-Board ROM Level II Block Diagram

**BIT TECHNIQUE: ON-BOARD ROM**

**CATEGORY: FLOW CHART / DESCRIPTION**

See Figure 3 for On-Board ROM Flow Chart.

1. A negative pulse "Test Initiate" signal is input to test control logic to begin test. The test begins as follows:

   - "BIT Mode" signal from control logic to multiplexer is activated
   - Normal inputs/outputs to/from CUT multiplexed out
   - Test Patterns (TP) from TP ROM input to CUT enabled
   - All resetable logic of CUT reset
   - Counters are preset
   - Pass/Fail Flip-Flop (FF) reset to Pass
   - Ensure CUT stable at falling edge of test clock. The test clock, while in BIT mode, increments the control logic counters which address the TP & GMR (Good Machine Response) ROMs simultaneously.

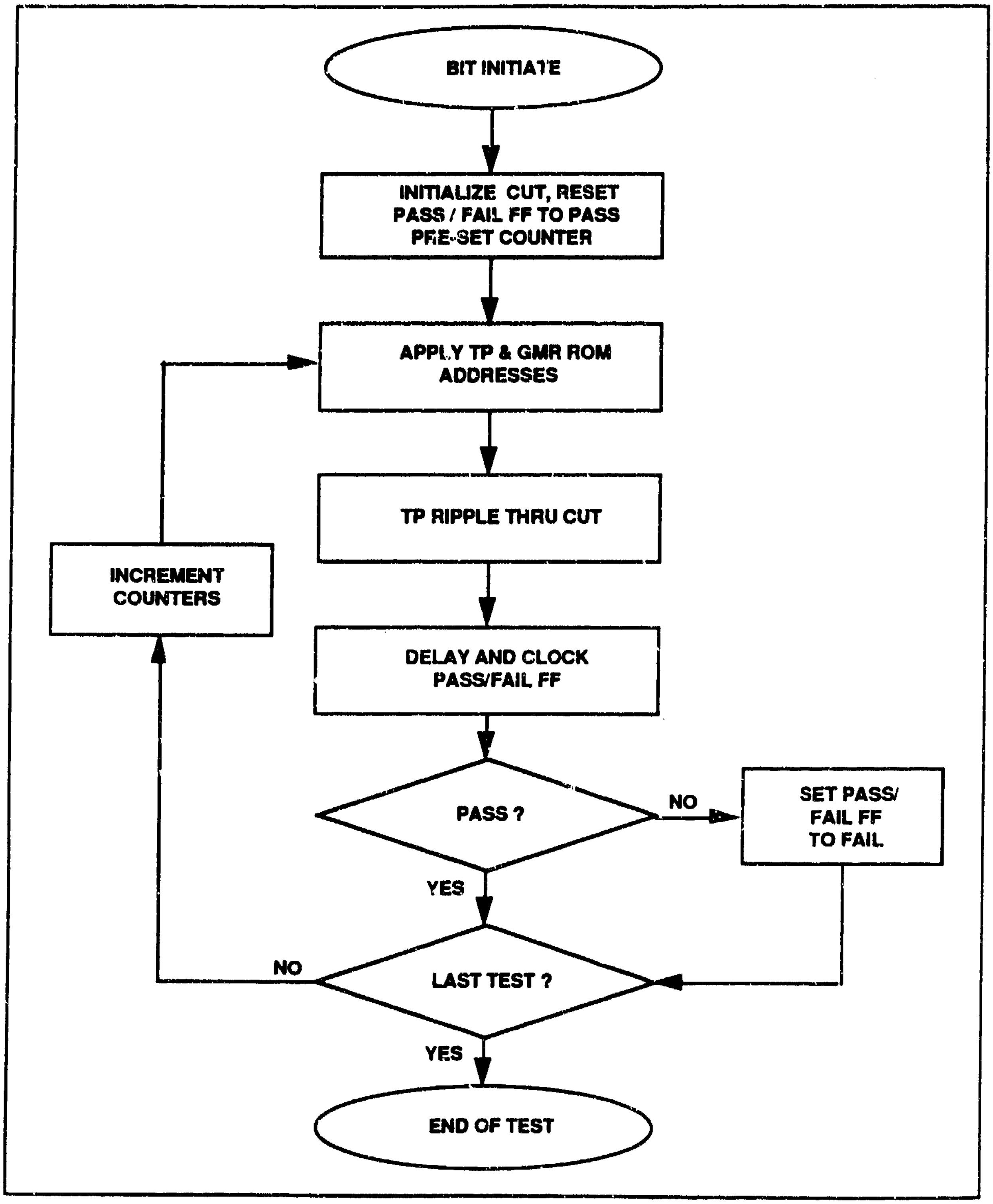
2. The TP ripples through the CUT.

3. After enough delay for a test pattern to establish a response at the CUT's output and be compared with the corresponding GMR, the inverted TEST CLOCK clocks the Pass/Fail Flip Flop to latch in the result.

4. A good machine at this time will have the GMR pattern identically compare with the CUT outputs. If not, the Pass/Fail FF will be set to "Fail" and will remain "Fail" until BIT is re-initiated.

5. If the address to the ROMs is the last address, then "End Of Test" control logic signal goes high, causing the BIT mode FF to reset and the system is out of BIT mode. The Pass/Fail FF will remain set to "Pass" if during the test it was never set to "Fail".

6. If not the last ROM address, increment the counter, apply the next TP and go to step 4. If at the last TP address, the RCO signal is triggered from most significant counter to end the test.


BIT INITIATE
INITIALIZE CUT, RESET PASS / FAIL FF TO PASS PRE-SET COUNTER
APPLY TP & GMR ROM ADDRESSES
TP RIPPLE THRU CUT
INCREMENT COUNTERS
DELAY AND CLOCK PASS/FAIL FF
PASS ?
NO
SET PASS/ FAIL FF TO FAIL
YES
NO
LAST TEST ?
YES
END OF TEST

274_844_51

Figure 3 BIT Sequence Flow Chart for On-Board ROM

## BIT TECHNIQUE: ON-BOARD ROM

## CATEGORY: ADVANTAGES

1. An understanding of the CUT can lead to a substantial percentage of fault detected with a few, predetermined test patterns.

2. A CUT with much sequential logic requires specific "Pairs" of test patterns applied in sequence. Although this presents a problem with random test pattern application, storing the test patterns in ROM so that they indeed do occur in pairs is done without difficulty with the On-Board ROM method. Also, specific GMRs (Good Machine Responses) can be generated by CAD circuit simulation with relative ease.

3. On-Board ROM Test Generation becomes competitive when compared to random pattern generation as the number of CUT inputs become large and/or number of patterns required becomes small. This is best understood by considering that the total number of binary patterns possible for a CUT with n inputs is $2^n$. If n=16; $2^n = 65{,}536$. If n=20; $2^n=1{,}048{,}576$. If n=24; $2^n=16{,}777{,}216$. Consider a 24-input CUT that can be adequately tested with 2,000 deterministic patterns (excluding all zero pattern). To be absolutely sure of providing all 2,000 test patterns one must cycle through 16,777,215 ($2^n$ -1) possible test patterns when using random pattern generator.

4. The control logic for the On-Board ROM technique is simple when compared to the random pattern generation test method which requires loading seed patterns and special test sequencing.

## CATEGORY: DISADVANTAGES

1. With the growing complexity of electronic circuitry being implemented on Line Replaceable Modules (LRM) of today, it is becoming more and more difficult for a test engineer to understand what he is testing, especially when under pressure to establish the test plan quickly. Without a true understanding of what is to be tested, it is nearly impossible to determine the test patterns that are necessary to completely and efficiently test the CUT.

2. When the number of test patterns required to obtain adequate fault coverage is large and/or the number of CUT inputs is small or can be partitioned into a few small number of input groups, then the real estate required for the On-Board ROM method becomes excessive when compared to the random pattern generation method.

3. Circuit design changes often require reprogramming the ROMs.

4. If the number of bus lines required to address the ROMs are excessive and/or the distance between the TP ROMs and the control logic, or between the GMR ROMs and the control logic is substantial, then Printed Circuit Board (PCB) real estate consumed is excessive and costly.

5. Memory allocated to either store test patterns or GMRs can never serve both test and function purposes as can shift registers used in Built in Logic Block Observers (BILBO), for example.

6. CUT and BIT logic throughput delays constrain maximum clock rate during test.

**BIT TECHNIQUE: ON-BOARD ROM**

**CATEGORY: ATTRIBUTES**

1. CONCURRENCY

- Nonconcurrent

2. TECHNOLOGY

- Digital

3. CUT MICROPROCESSOR REQUIRED ?

- No

4. CUT INTERNAL DESIGN REQUIRED ?

- No

5. AREA PENALTY

- Increases with CUT complexity

6. WEIGHT PENALTY

- Increases with CUT complexity

7. POWER PENALTY

- Increases with CUT complexity. To minimize power penalty, use ROMs that have power down mode.

8. TIMING PENALTY

- Test Time = Number of Test Patterns divided by TP application rate.
- Throughput Delay = Additional delay of MUX

**BIT TECHNIQUE: ON-BOARD ROM**

**CATEGORY: ATTRIBUTES, Contd**

9. RELIABILITY IMPACT

- Proportional to Area Penalty if similar technology is used for Built-In-Test Equipment (BITE) as for CUT.
- May have to distinguish BITE failures that only effect BITE vs BITE failures that effect CUT.
- Computer-Aided Design (CAD) System may have software package for reliability calculation.

10. CONCEPTUAL COMPLEXITY

- Straightforward

11. HARDWARE/SOFTWARE/FIRMWARE

- Test Patterns are stored as Firmware in ROM

12. DESIGN COST

- Use standard estimating procedure based on number of chips.
- Engineering time to create Test Patterns
- Debug time to verify proper operation.

13. MEMORY REQUIREMENTS

- Proportional to number of test patterns and type of ROM chip.

14. BIT CIRCUITRY SELF-TESTABLE ?

- Can do checksum on ROMs with added hardware.

15. STAND-ALONE (SELF-CONTAINED) BIT?

- Yes

16. NOTES

- None

**BIT TECHNIQUE: ON-BOARD ROM**

**CATEGORY: DEFAULT DESIGN**

This section describes the sequence of events for the ONBOARD ROM BIT technique default design. Refer to Figure 4 for the Default Design schematic.

1. At the beginning of the test, the TEST_INITIATE signal goes low and triggers the following:

   a. At the falling edge of the TEST_INITIATE signal, U2A test control flip-flop outputs high to enable TEST_CLOCK signal to activate counters.

   b. While the TEST_INITIATE signal holds low, the next rising edge of the TEST_CLOCK signal loads in the preset value (beginning address) for the synchronous 4-bit binary counters U4, U5, U6, and U7.

   c. Disables normal inputs of CUT. Configures U9 and U10 MUX to select inputs from Test Pattern ROM to CUT.

   d. Disables U13A and U13B octal buffers and places the outputs of the CUT into high impedance.

   e. Resets all resetable logics in the CUT.

   f. Presets the Pass/Fail flip-flop U2B to PASS (use _Q output).

2. After the TEST_INITIATE signal goes back to high, the TEST_CLOCK signal starts incrementing counters U4, U5, U6, and U7. While the counters are driven directly by TEST_CLOCK signal, the result of the comparison of Good Machine Response and test response is captured at the falling edge of the TEST_CLOCK signal to ensure enough delay time for the test pattern to ripple through the CUT before it reaches the 8-bit identity comparator.

3. At the end of the test, U7/RCO outputs a high and triggers the following:

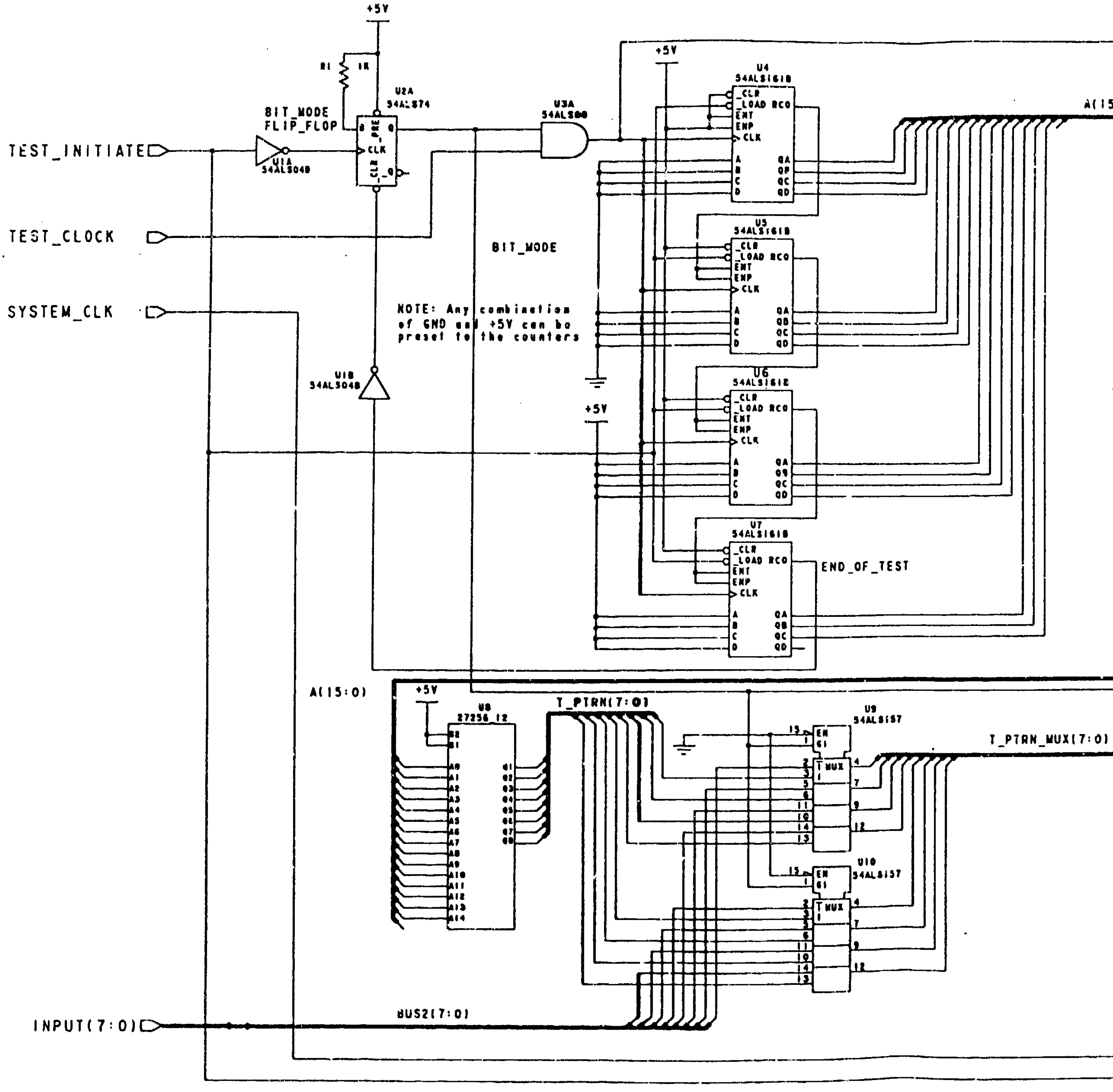
   a. Clears U2A test control flip-flop and disconnects the TEST_CLOCK signal to counters.

   b. Enables normal inputs of CUT. Configures U9 and U10 MUX to select inputs from normal inputs to CUT.

   c. Enables U13A and U13B octal buffers for normal CUT outputs.

   d. Any failure occuring during the test is captured by the Pass/Fail flip-flop U2B.

4. The circuit resumes normal operation and it will go into test mode when the TEST_INITIATE signal goes low again.


TEST_INITIATE
TEST_CLOCK
SYSTEM_CLK
INPUT(7:0)
+5V
R1
1K
U2A
54ALS74
BIT_MODE
FLIP_FLOP
PRE
CLK
CLR
Q
U1A
54ALS04B
U3A
54ALS08
BIT_MODE
U1B
54ALS04B
NOTE: Any combination of GND and +5V can be preset to the counters
U4
54ALS161B
U5
54ALS161B
U6
54ALS161B
U7
54ALS161B
_CLR
_LOAD
RCO
ENT
ENP
CLK
A
B
C
D
QA
QB
QC
QD
END_OF_TEST
A(15
A(15:0)
U8
27256_12
E2
E1
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
O1
O2
O3
O4
O5
O6
O7
O8
T_PTRN(7:0)
U9
54ALS157
U10
54ALS157
EN
G1
MUX
T_PTRN_MUX(7:0)
BUS2(7:0)

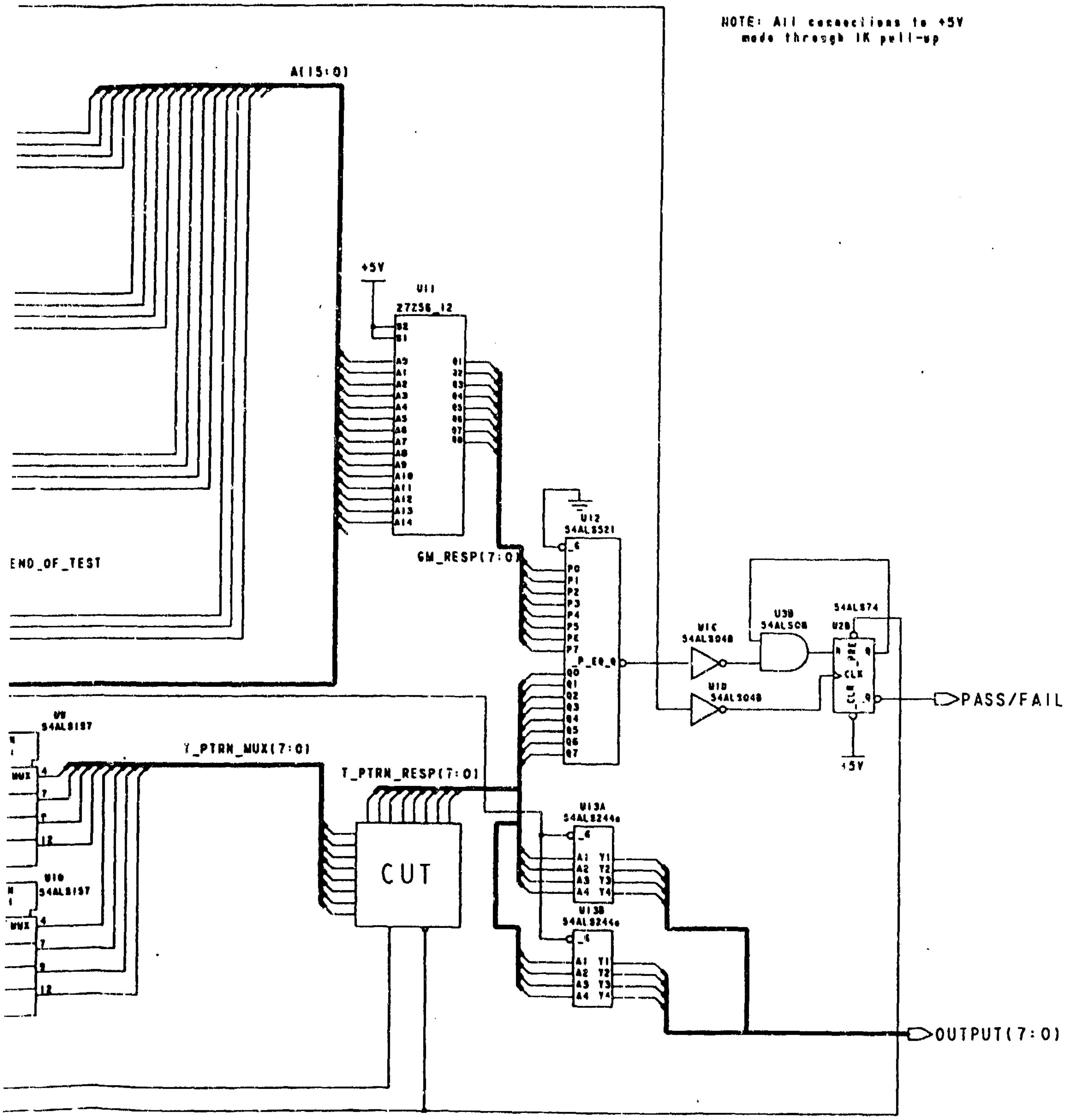


Figure 4 On-Board ROM – Default Design

## BIT TECHNIQUE: ON-BOARD ROM

## CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM

Figure 5 shows the BIT Technique Insertion Diagram for this BIT technique.

BIT ELEMENT 1

Connect the MUX output to the CUT inputs.

BIT ELEMENT 2

Connect normal LRM (Line Replaceable Module) inputs and the TP (Test Pattern) Rom outputs to the MUX inputs.

BIT ELEMENT 3

Connect the CUT outputs to the BUFFER inputs

BIT ELEMENT 4

Connect the CUT outputs to the COMPARATOR inputs.

BIT ELEMENT 5

Connect the GMR (Good Machine Response) ROM outputs to the COMPARATOR inputs.

BIT ELEMENT 6

Connect the COMPARATOR outputs to the PASS/FAIL FLIP FLOP.

BIT ELEMENT 7

1. Connect the counter outputs to the address lines of the TP & GMR ROMs.
2. Tie counters preset inputs to load the beginning address of BIT routine.
3. Connect TEST INITIATE to BIT mode FLIP FLOP and PASS/FAIL FLIP FLOP.
4. Use TEST CLOCK to clock counters and invert the TEST CLOCK to clock PASS/FAIL FF. The falling edge of the TEST CLOCK signal triggers the PASS/FAIL FF after the total settling time of the circuit.
5. Set up RCO signal from most significant counter to indicate END OF BIT, disable BIT circuitry and return CUT to normal operational mode.

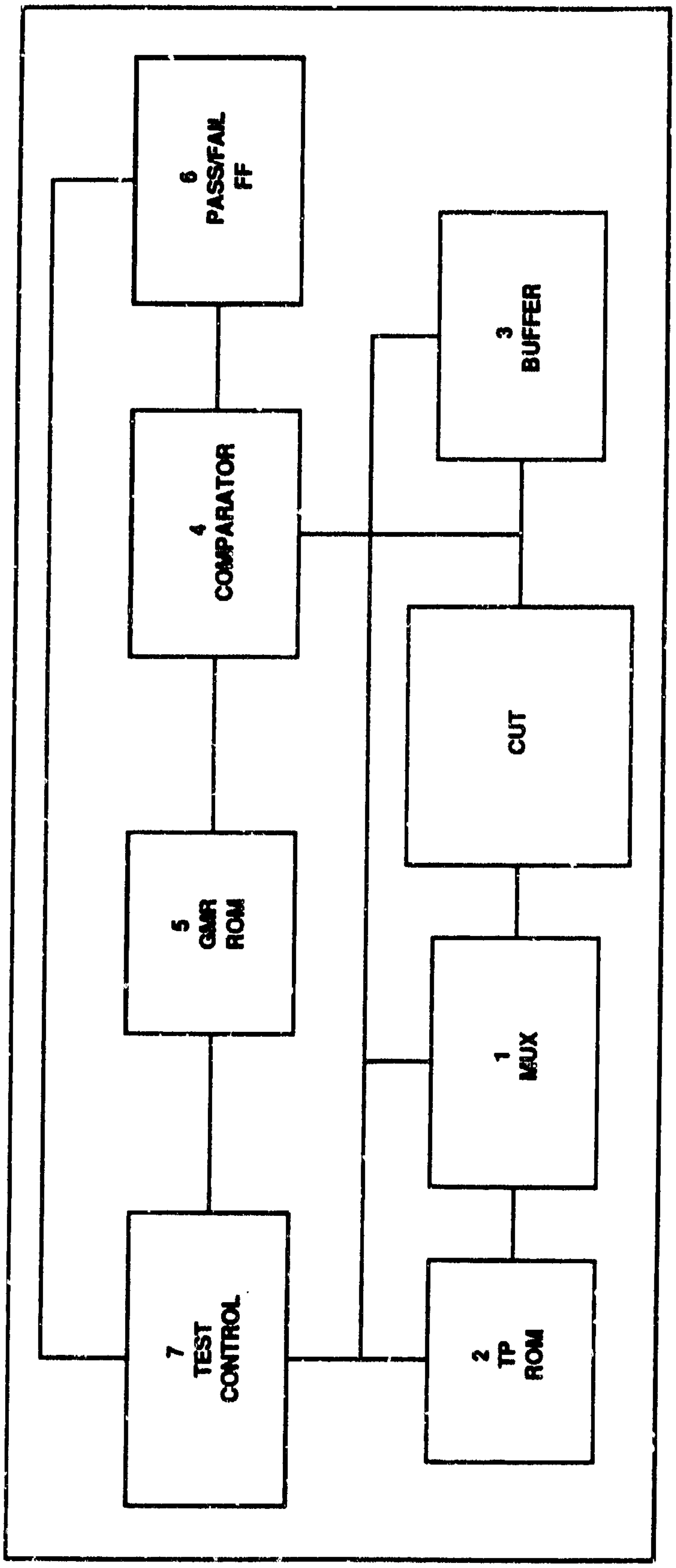


Figure 5 BIT Technique Insertion Diagram for On-Board ROM

## BIT TECHNIQUE: ON-BOARD ROM

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | Number of CUT inputs | none |
| $v_2$. | Number of CUT outputs to test | none |
| $v_3$. | Maximum propagation delay through CUT | ns |
| $v_4$. | CUT initialization time | sec |
| $v_5$. | #Test patterns for On-Board ROM BIT technique | #patterns |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 54ALS04B | INVERTER | 4 |
| 2 | 54ALS08 | AND GATES | 2 |
| 3 | 54ALS74A | FLIP-FLOP | 2 |
| 4 | 54ALS157 | SELECTOR/MUX | ceil ( $v_1$ / b ) |
| 5 | 54ALS161B | COUNTER | ceil ( log2($v_5$) / b ) |
| 6 | 54ALS244A | BUFFER | ceil ( $v_2$ / b ) |
| 7 | 54ALS521 | COMPARATOR | ceil ( $v_2$ / b ) |
| 8 | 27C256-12 | PROM | ceil ( $v_5$ / m ) ( ceil($v_1$/b)+ceil($v_2$/b) |
| 9 | RNR55 | RESISTOR | 1 |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i \quad = \quad \text{ceil} ( u_i / upp_i )$$

**Explanation of symbols used:**

| | | |
|---|---|---|
| $n_i$ | = | Number of components (physical packages) required for i-th part |
| $u_i$ | = | Number of units (CAD symbols) required for i-th part |
| $upp_i$ | = | Number of units/package for i-th part (from Table 4.0) |
| ceil | = | Round up to nearest integer (e.g. 5.3 is rounded to 6) |
| b | = | Number of data bits for part (from Table 4.0) |
| m | = | Number of memory locations for part = $2^a$ |
| a | = | Number of address bits for part (from Table 4.0) |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |

**BIT TECHNIQUE: ON-BOARD ROM**

**CATEGORY: PENALTY EQUATIONS**

**AREA (sq.in.)** = Sum ( $n_i * a_i$ ) + 15% for traces

**WEIGHT (gms)** = Sum ( $n_i * w_i$ ) + 10% for solder

**POWER (mW)** = Sum ( $n_i * p_i$ )

**TEST TIME (ns)** = Max ($v_4 * 10^9$, 50) + ( $v_5$ * 2M * (tMUX + tCOMP + $v_3$) ) + tFF

**DELAY (ns)** = tMUX + tBUFFER

**Explanation of Symbols Used.**

Sum = Sum over parts in BIT Technique Parts Table (i=1 to 9)<br>
$n_i$ = Number of component packages for i-th part in Parts Table
$n_i$ is calculated according to Component Determination Equations

$a_i$ = Area in sq.in. of i-th part in Parts Table

$p_i$ = Power dissipation in mW (typical) for i-th part in Parts Table

$w_i$ = Weight in grams of i-th part in Parts Table

$v_i$ = User-supplied value for i-th variable (see Variable Definitions)

M = Design margin = 1.2

tMUX = Max delay for SELECTOR/MUX from Table 4.0

tCOMP = Max delay for COMPARATOR from Table 4.0

tFF = Max delay for FLIP-FLOP from Table 4.0

tBUFFER = Max delay for BUFFER from Table 4.0

**CATEGORY: BIBLIOGRAPHY**

None required.

# MICROPROCESSOR BIT

# BIT TECHNIQUE DATA PACKAGE

## CATEGORY: SHORT DESCRIPTION / DIAGRAMS

Microprocessor Built-In-Test (BIT) is accomplished using functional fault models which comprehensively and efficiently test the microprocessor. To implement this method, some test program memory and the built-in intelligence of the microprocessor are required. An optional external test module may also be used, depending on the circuit being tested. The external test module is a circuit controlled by the Central Processing Unit (CPU) and is used to control or initiate microprocessor peripheral control devices which are located on the microprocessor module.

Microprocessor BIT is done in stages. Each subsequent stage builds upon the successful completion of prior stages. These stages are performed in the specific order shown below:

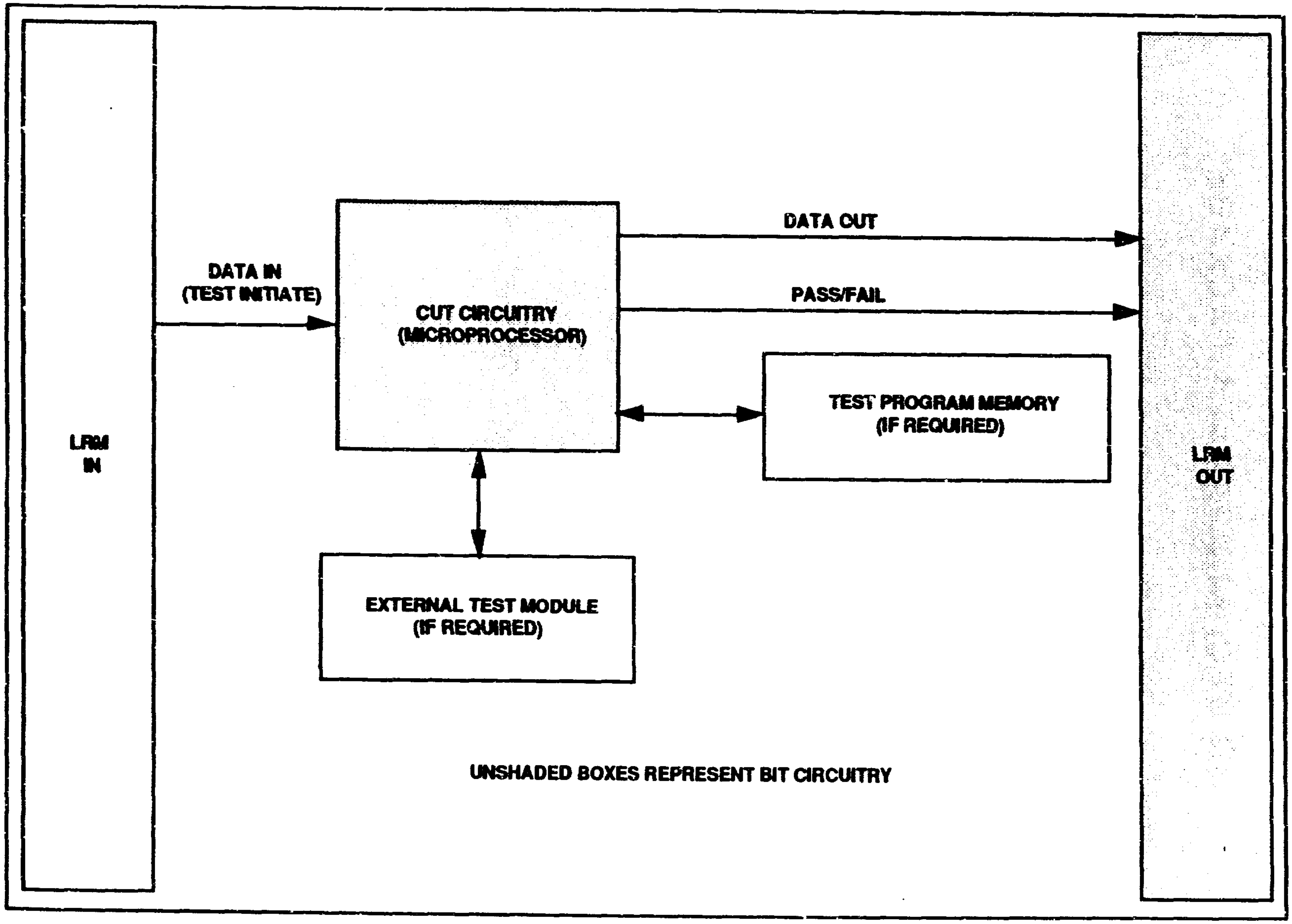
- Core Instruction tests
- Read Register Instruction tests
- Memory tests
- Addressing Modes tests
- Instruction Execution tests
- Instruction Sequence tests
- I/O Peripheral Controller tests

In addition to the microprocessor, the external test module may be used to implement a test. It is used in the following manner:

- Verify CPU is operating properly (see above list).
- Set up on-chip peripheral controllers to external control mode.
- Use the external test module to set up external on-chip peripheral controller requests.
- Return on-chip peripheral controllers to operational mode

Normally, Microprocessor BIT is executed at the operating speed of the microprocessor.

Figures 1 and 2 show the Level I and II Block Diagrams for this BIT technique.



274_844_109

Figure 1 Level I Block Diagram for Microprocessor BIT

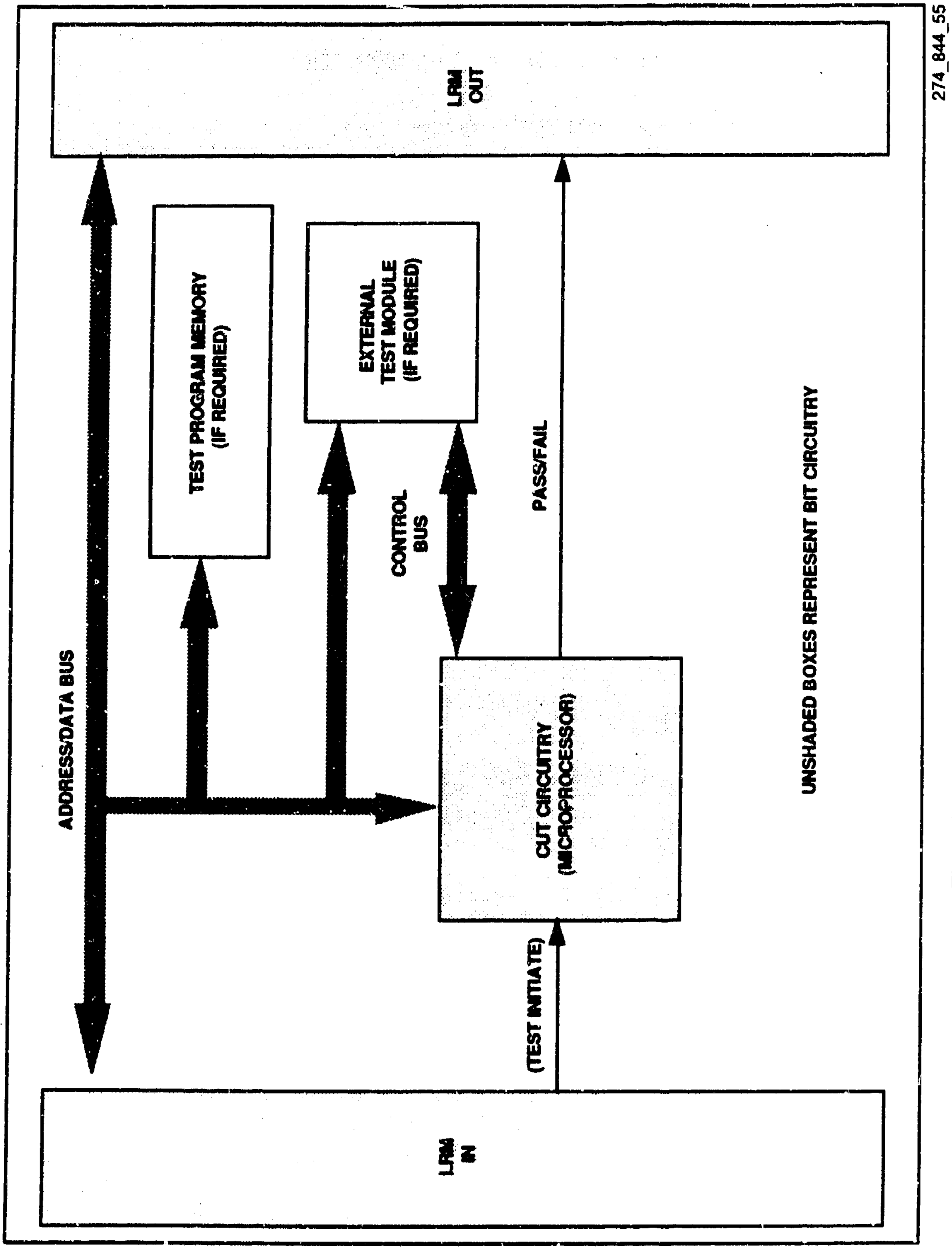

LRM IN
LRM OUT
ADDRESS/DATA BUS
TEST PROGRAM MEMORY (IF REQUIRED)
EXTERNAL TEST MODULE (IF REQUIRED)
CONTROL BUS
PASS/FAIL
(TEST INITIATE)
CUT CIRCUITRY (MICROPROCESSOR)
UNSHADED BOXES REPRESENT BIT CIRCUITRY
274_844_55

Figure 2 Level II Block Diagram for Microprocessor BIT

## BIT TECHNIQUE: MICROPROCESSOR BIT

## CATEGORY: FLOW CHART / DESCRIPTION

Figure 3 provides a Flow Chart for this BIT technique.

1. Generate an initiate BIT signal.

2. Set Pass/Fail Output to PASS.

3. Execute a procedure which verifies the proper operation of the MOVE, COMPARE, and BRANCH instructions. These instructions are typically defined as follows:

   - MOV a, Ri: Load register Ri with the contents of memory location a.
   - CMP Ri, Rj: Compare the contents of Ri to Rj and set the Z bit if Ri = Rj.
   - BEQ b: If the Z bit of the Status Register (SR) is set, then branch to location b.

These instructions must be operational before any further testing can proceed because they are a kernel which enables testing the execution of further instructions in the instruction repertoire.

4. If a fault is detected in the Core Instructions test,set the Pass/Fail Output to FAIL and terminate testing. If no faults are detected, proceed to the Read Register instruction tests.

5. Execute a procedure which verifies proper execution of the Read Register Instructions of the microprocessor. The procedure verifies that the proper data is read and checks for simple faults.

6. If a fault is detected, set Pass/Fail Output to FAIL. If no faults are detected, testing will proceed to the Memory test.

7. Execute a procedure which verifies proper operation of the memory chips.

8. If a fault is detected, set Pass/Fail Output to FAIL. If no faults are detected, proceed to the addressing modes tests.

9. Execute a procedure which verifies proper loading of registers in all the addressing modes of the microprocessor. This verifies that all addressing modes are functional.

**BIT TECHNIQUE: MICROPROCESSOR BIT**

**CATEGORY: FLOW CHART DESCRIPTION, Contd**

10. If a fault is detected, set Pass/Fail Output to FAIL. If no faults are detected, proceed to the Instruction Execution test.

11. Execute a procedure which verifies that the Instruction Execution process is functional. This is accomplished by loading the registers with codewords, executing an Instruction set, and verifying the proper content of the registers.

12. If a fault is detected, set Pass/Fail Output to FAIL. If no faults are detected, proceed to the Instruction Sequence test.

13. Execute a procedure in which all possible ordered pairs of instructions are tested. Ordered pairs of instructions are defined as instructions which are commonly used together. Check for the following faults:

    - No data dependence (the sequence fault is independent of the operands used with the instructions).
    - Pairwise instruction sequence dependence.

14. If a fault is detected, set Pass/Fail Output to FAIL. If no faults are detected, testing will proceed to the Integrated Controller tests.

15. In general, the fault model for any on-chip peripheral controller is as follows:

    - Registers belonging to the peripheral control device have stuck-at faults. The result of these faults will be incorrect, or there will be no execution of the device function.
    - Faults in decoders of the peripheral control device cause incorrect, or no selection of peripheral control registers.
    - Faults in the control logic of the peripheral cause incorrect, or no, execution of the controller function.
    - A fault in the on-chip peripheral may cause a side effect in other areas of the microprocessor which may be detected in its readable registers.

16. If a fault is detected,set Pass/Fail Output to FAIL. If no faults are detected, the Microprocessor BIT passed.

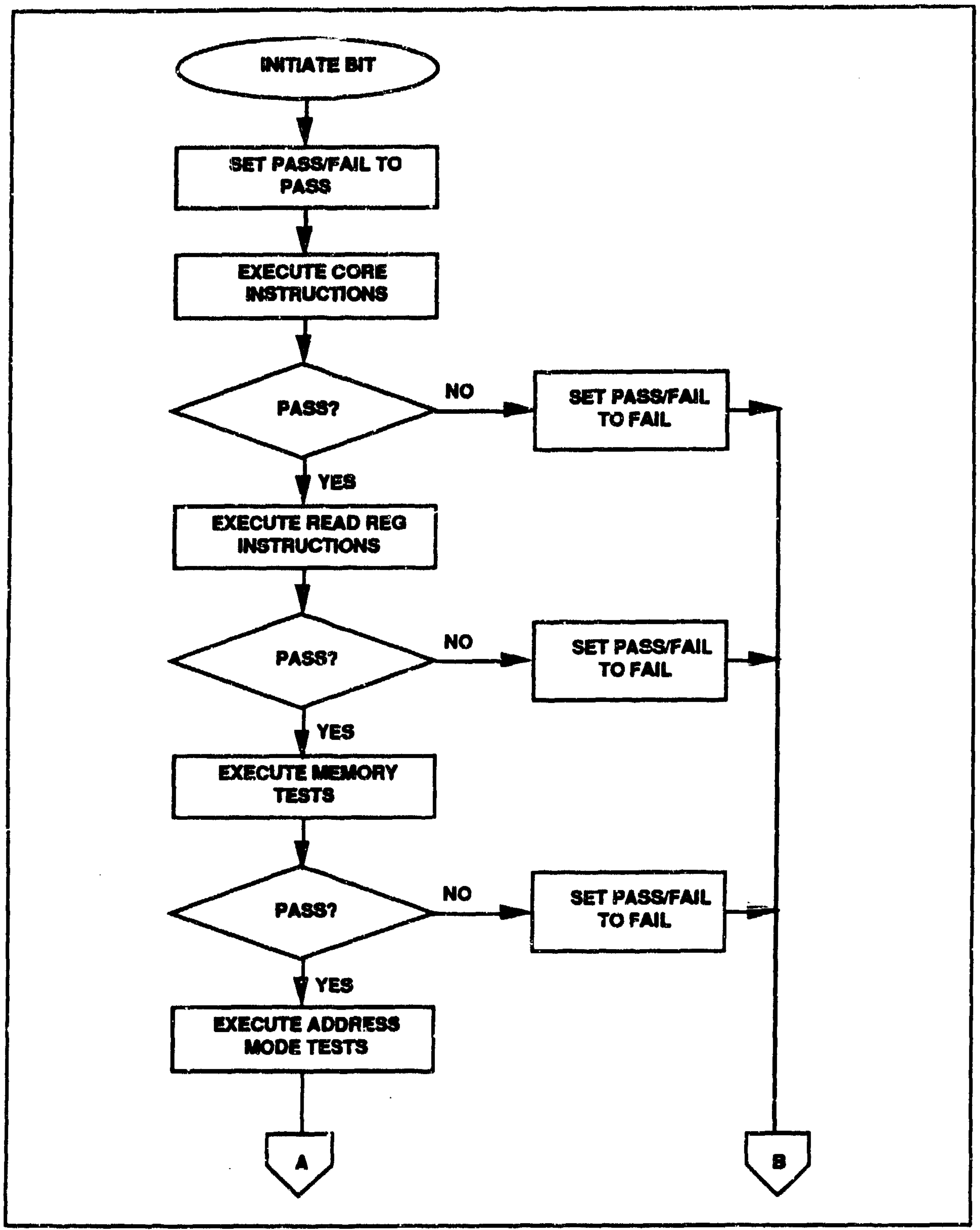
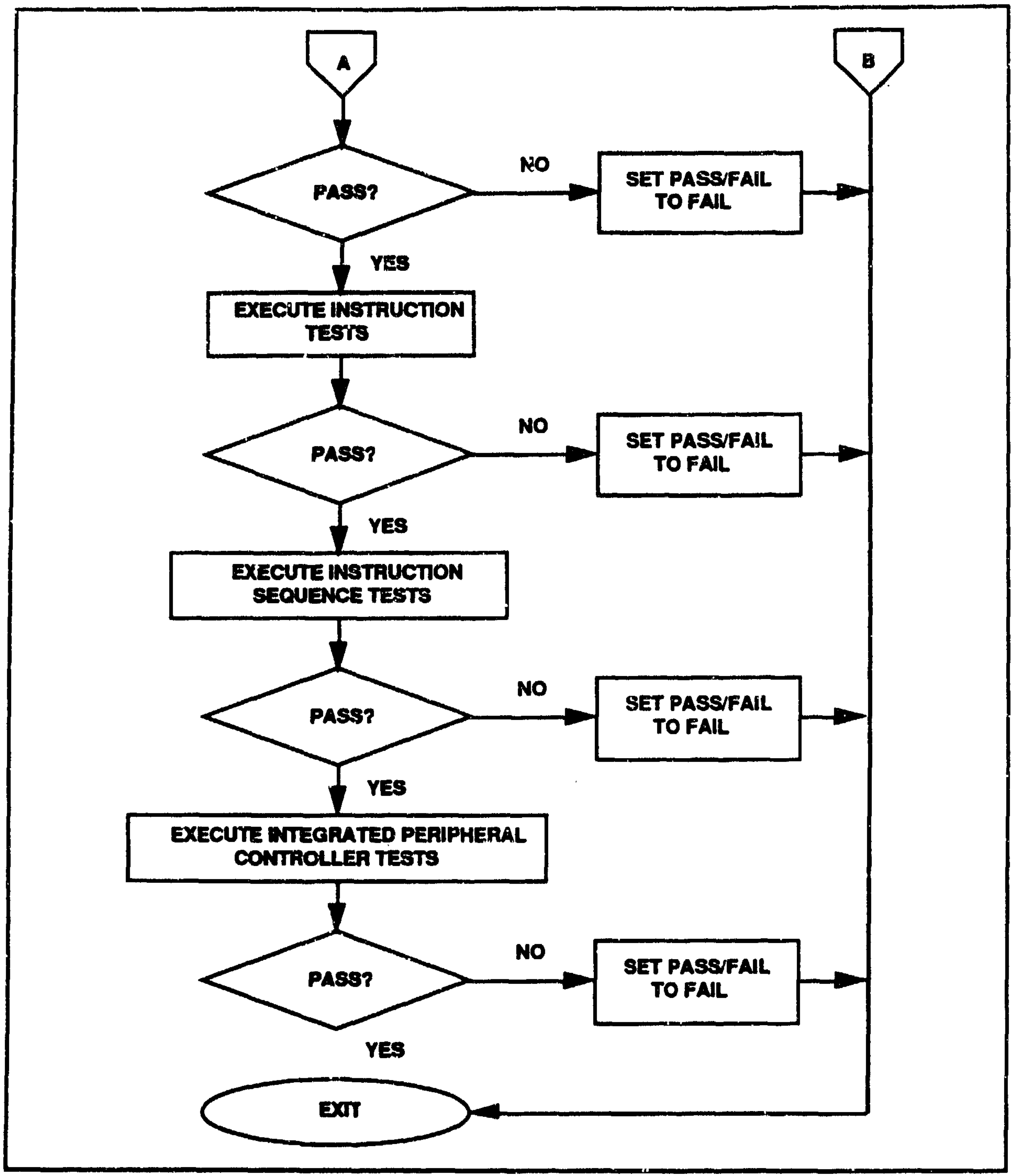


Figure 3 BIT Sequence Flow Chart for Microprocessor BIT (Sheet 1 of 2)



274_844_54

Figure 3 BIT Sequence Flow Chart for Microprocessor BIT (Sheet 2 of 2)

## BIT TECHNIQUE: MICROPROCESSOR BIT

## CATEGORY: ADVANTAGES

Microprocessor BIT technique provides the following advantages to the circuit designer:

1. The real estate penalty will be minimal, basically requiring Read Only Memory (ROM) locations which may already be available if there is spare ROM space after the design is complete. If an external test module is required, the real estate penalty will be slightly increased.

2. Most of the tests are executed at the operating speed of microprocessor.

3. Monitoring of test results is carried out by the microprocessor itself.

**BIT TECHNIQUE: MICROPROCESSOR BIT**

**CATEGORY: DISADVANTAGES**

Microprocessor BIT technique poses the following disadvantages to the circuit designer:

1. The test memory requirement can be large, and is dependent on the following factors:

   - Microprocessor characteristics
   - Thoroughness of the tests
   - Optimization of the test code for fast execution

2. Most of the test code must be written in assembly language or machine code. Assembly and machine codes are not as readable as high order languages.

**BIT TECHNIQUE: MICROPROCESSOR BIT**

**CATEGORY: ATTRIBUTES**

1. CONCURRENCY

- Concurrent – BIT must be run infrequently during normal operation to maintain concurrency.

2. TECHNOLOGY

- Digital

3. CUT MICROPROCESSOR REQUIRED

4. CUT INTERNAL DESIGN REQUIRED?

- No, unless additional ROM and/or external test module are needed.

5. AREA PENALTY

- The number of memory chips needed is proportional to the total memory requirement of the self-test program used. If it will not fit in existingmemory, penalty will be accrued consisting of:

a. Several registers

b. Control logic circuitry

6. WEIGHT PENALTY

- Weight increases as the number of additional memory chips increases.

7. POWER PENALTY

- Proportional to the number of added memory chips.
- If an External Test Module and/or memory is required, an additional power penalty will be accrued.

**BIT TECHNIQUE: MICROPROCESSOR BIT**

**CATEGORY: ATTRIBUTES, Contd**

8. TIMING PENALTIES

- Test time is a function of clock frequency and test program length
- Throughput delay is 6 ns.

9. RELIABILITY IMPACT

- Failure rate increases with the addition of memory chips.
- If an External Test Module is required, an additional reliability penalty will be accrued.

10. CONCEPTUAL COMPLEXITY

- Circuit design is straightforward.
- Some assembly language programming is required

11. HARDWARE/SOFTWARE/FIRMWARE

- Some hardware required if additional ROM or external test module are required
- Assembly and/or machine language code are required
- Test software may be either ROM resident or loaded prior to execution

12. DESIGN COST

- All components used are readily available at low cost.
- Software development time of the BIT programs stored in memory increases with the complexity and thoroughness of the tests used.
- Hardware design and debug is minimal.

13. MEMORY REQUIREMENTS

- Proportional to size of test software

14. BIT CIRCUITRY SELF-TESTABLE?

- Yes

15. STAND-ALONE (SELF-CONTAINED) BIT?

- Yes

## BIT TECHNIQUE: MICROPROCESSOR BIT

## CATEGORY: DEFAULT DESIGN

This section describes the default design for the Microprocessor BIT technique. Refer to Figure 3 for the Level II Block Diagram for this BIT technique and Figure 4 for the Default Design schematic.

In many cases, Microprocessor BIT can be accomplished by using the available ROM to store the BIT software. Additional hardware is needed if either the BIT software will not fit in the existing ROM, or if the designer wishes to test the microprocessor's hardware connections. The external microprocessor pin being tested is an interrupt. Although the interrupt function can be tested using software, the actual hardware pin function cannot be tested without external stimulus. Figure 4 shows how additional ROM can be added, and a method to test microprocessor inputs.

The additional ROM is shown as U3. It is wired to the microprocessor in the same manner as any existing ROM, only mapped to a different location. The 8255A is mapped to the base address 8000H by connecting its chip select to the inverted most significant bit of the 16-bit address bus. The outputs of the 8255 drive a 54ALS157 multiplexer. During normal operation, the existing system interrupt is routed through the multiplexer to the P3.2 pin on the microprocessor. Under software control, the 8255 can use the multiplexer to route an 8255 output to the microprocessor's interrupt input pin (P3.2), thus allowing the BIT software to exercise the port and determine if it is working properly. After the test, the BIT software would return the configuration to normal.

NOTE: All conne
through 1

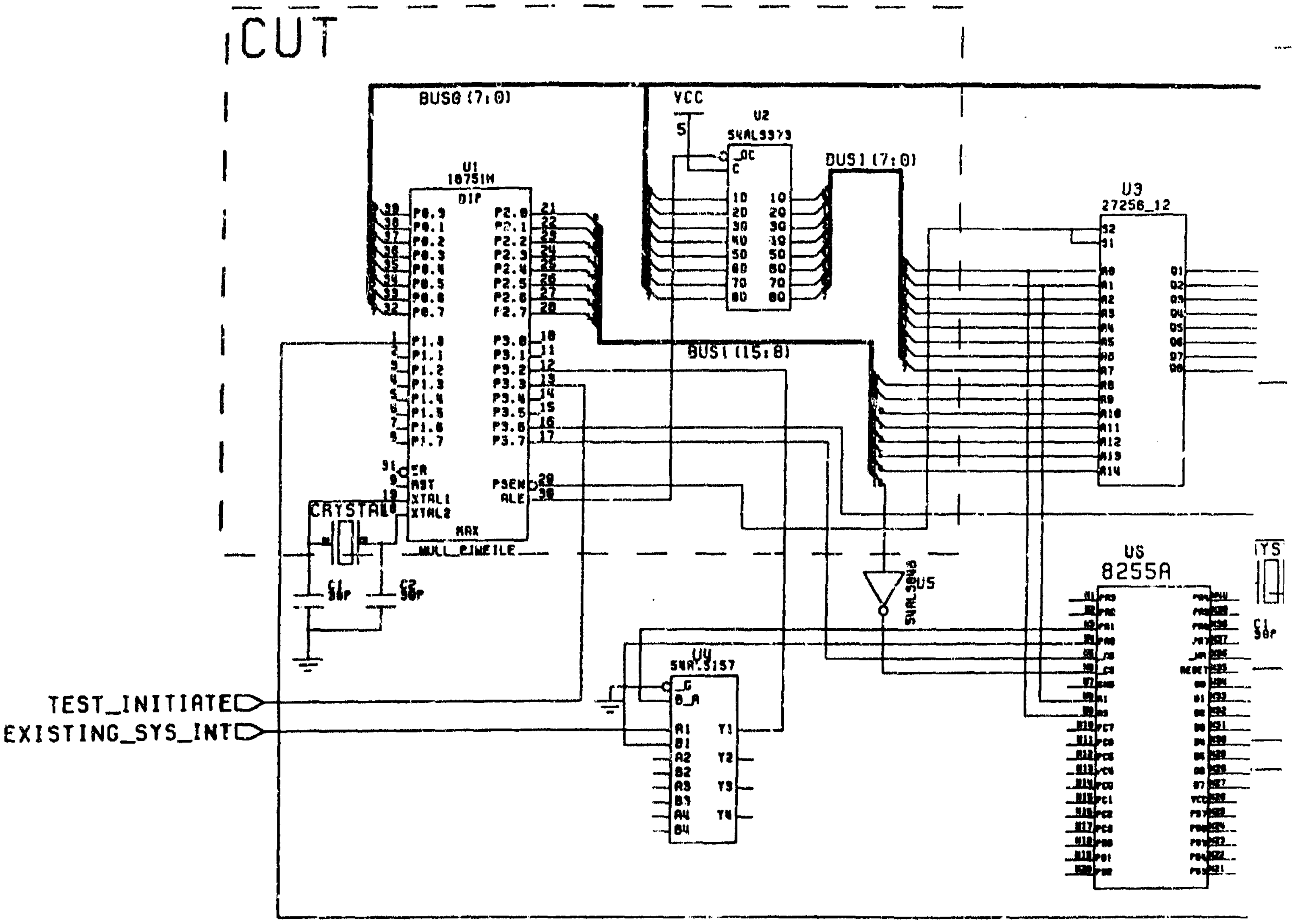


Figure 4 Microprocessor BIT – Default Desig

1

NOTE: All connections to VCC made through 1K resistor pull up

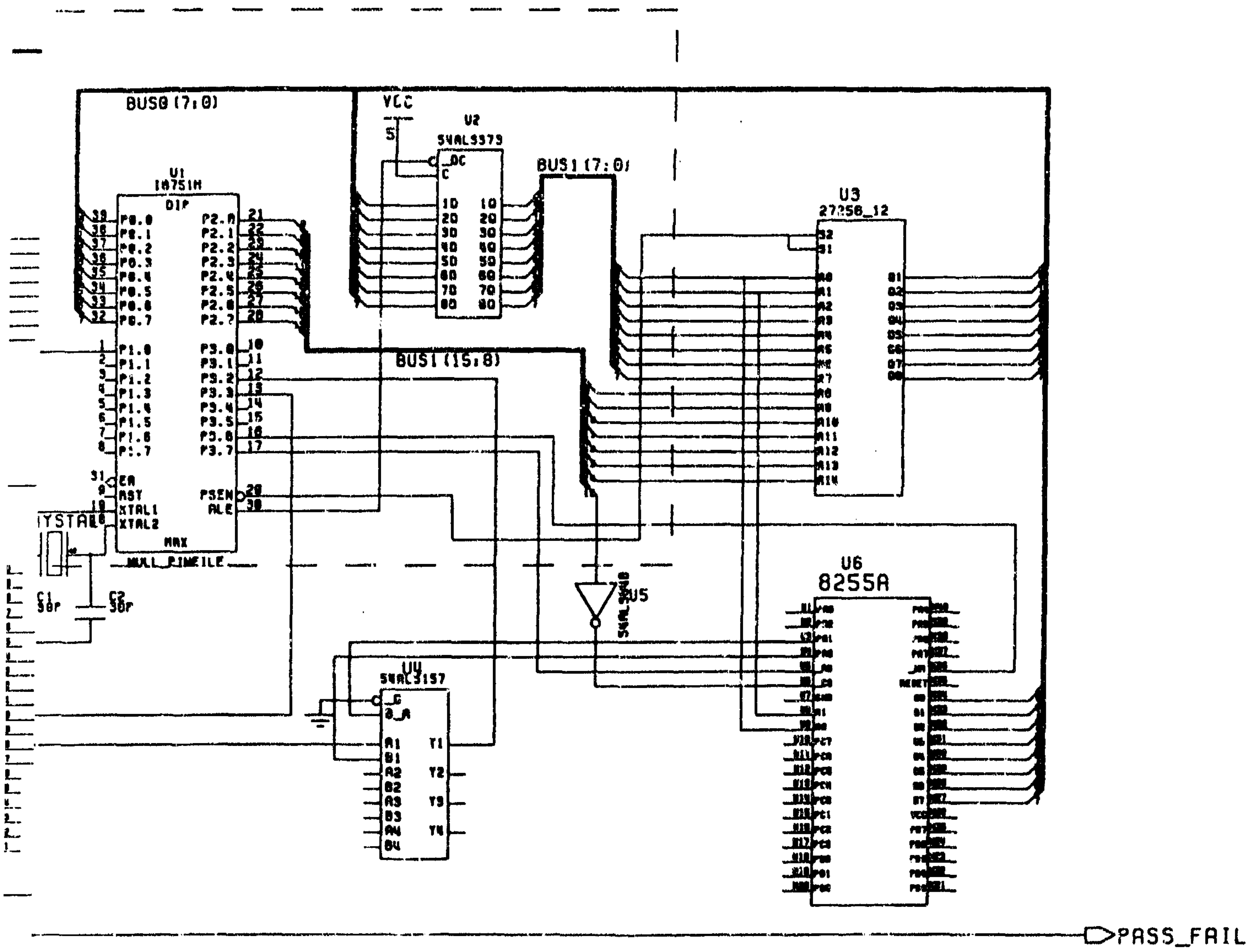


Figure 4 Microprocessor BIT – Default Design

6

## BIT TECHNIQUE: MICROPROCESSOR BIT

## CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM

Refer to Figure 5 for the BIT Technique Insertion Diagram.

1. Test Program Memory

   If additional ROM is required, connect the data, address, and enable lines of the 47256 ROM to the CUT data, address, and memory chip decoder circuit.

2. External Test Module

   If an external test module is required, insert the 8255A using the CUT I/O port decoder logic. Determine which external stimulus (i.e., interrupt, I/O port) lines need to be tested. Connect the 54ALS157 multiplexer Y output(s) to the microprocessor input(s) to be tested. Connect the multiplexer A input(s) to the existing stimulus signal(s).

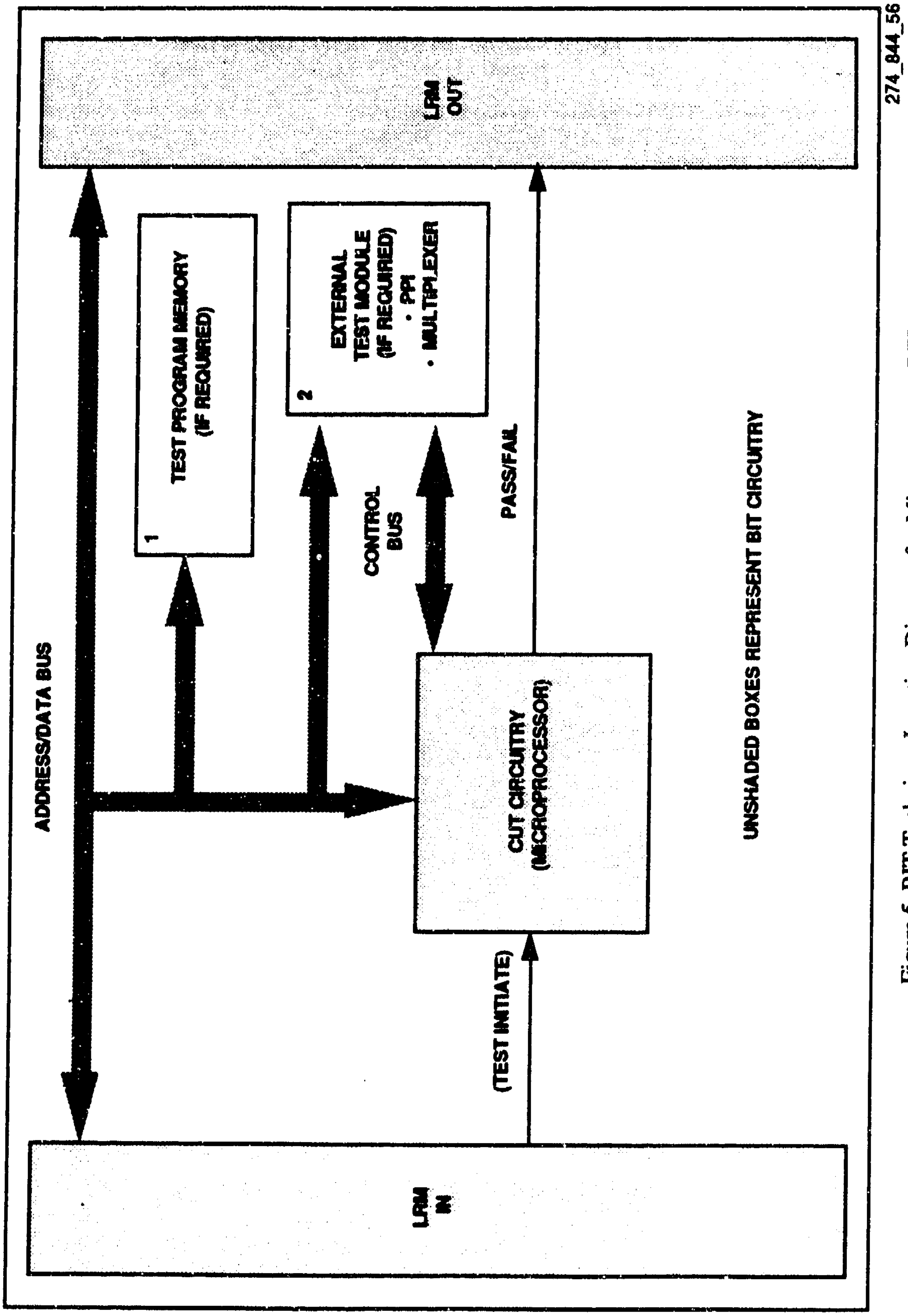

274_844_56
LRM OUT
ADDRESS/DATA BUS
1
TEST PROGRAM MEMORY (IF REQUIRED)
2
EXTERNAL TEST MODULE (IF REQUIRED)
· PPI
· MULTIPLEXER
CONTROL BUS
PASS/FAIL
CUT CIRCUITRY (MICROPROCESSOR)
UNSHADED BOXES REPRESENT BIT CIRCUITRY
(TEST INITIATE)
LRM IN

Figure 5 BIT Technique Insertion Diagram for Microprocessor BIT

## BIT TECHNIQUE: MICROPROCESSOR BIT

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | #Bytes for Microprocessor BIT S/W in add'l ROM | bytes |
| $v_2$. | Data bus width | bytes |
| $v_3$. | #Peripheral functions to test | functions |
| $v_4$. | Microprocessor execution speed | instructions/sec |
| $v_5$. | #Dynamic instructions for Microprocessor BIT S/W | instructions |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 54ALS04B | INVERTER | 1 |
| 2 | 54ALS157 | SELECTOR/MUX | ceil ( $v_3$ / b ) |
| 3 | 27C256-12 | PROM | ceil ( $8v_2$ / b ) * ceil(($v_1$/$v_2$)/m) |
| 4 | 8255A | PERIPHERAL I/F | ceil ( $v_3$ / b ) |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i = \text{ceil} ( u_i / \text{upp}_i )$$

**Explanation of symbols used:**

$n_i$ = Number of components (physical packages) required for i-th part
$u_i$ = Number of units (CAD symbols) required for i-th part
$\text{upp}_i$ = Number of units/package for i-th part (from Table 4.0)
ceil = Round up to nearest integer (e.g. 5.3 is rounded to 6)
b = Number of data bits for part (from Table 4.0)
m = Number of memory locations for part = $2^a$
a = Number of address bits for part (from Table 4.0)
$v_i$ = User-supplied value for i-th variable (see Variable Definitions)

**BIT TECHNIQUE: MICROPROCESSOR BIT**

**CATEGORY: PENALTY EQUATIONS**

| | | |
|---|---|---|
| **AREA (sq.in.)** | = | Sum ( $n_i$ * $a_i$ ) + 15% for traces |
| **WEIGHT (gms)** | = | Sum ( $n_i$ * $w_i$ ) + 10% for solder |
| **POWER (mW)** | = | Sum ( $n_i$ * $p_i$ ) |
| **TEST TIME (ns)** | = | 0 (This technique has been classified as Concurrent) |
| **DELAY (ns)** | = | 6 |

**Explanation of Symbols Used.**

| | | |
|---|---|---|
| Sum | = | Sum over parts in BIT Technique Parts Table (i=1 to 4) |
| $n_i$ | = | Number of component packages for i-th part in Parts Table<br>$n_i$ is calculated according to Component Determination Equations |
| $a_i$ | = | Area in sq.in. of i-th part in Parts Table |
| $p_i$ | = | Power dissipation in mW (typical) for i-th part in Parts Table |
| $w_i$ | = | Weight in grams of i-th part in Parts Table |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |

**CATEGORY: BIBLIOGRAPHY**

None required.

# MICRODIAGNOSTICS

# BIT TECHNIQUE DATA PACKAGE

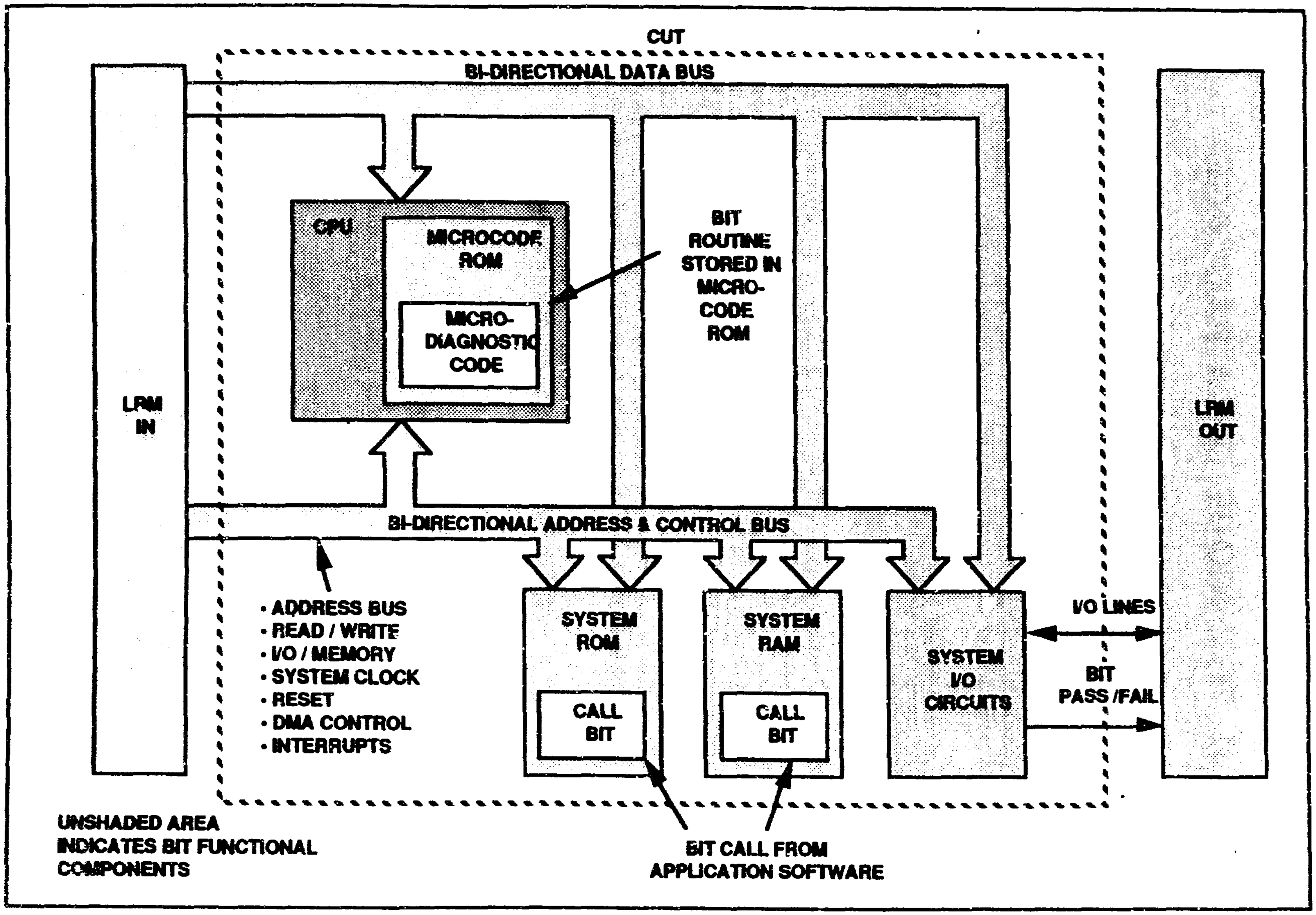
## CATEGORY: SHORT DESCRIPTION / DIAGRAMS

Microdiagnostics is a diagnostic BIT technique that is implemented by microprogramming on a microcode level. BIT implemented at the microcode level in a microprogramming environment allows hardware and software testing without the need for a hardware intensive approach compared to an application software level BIT running out of RAM or ROM.

The technique involves partitioning an area of the microcode ROM to support a small BIT routine which is executed by a macro instruction. For the purposes of this document, the macroinstruction will be called RUNBIT. When the opcode for RUNBIT is encountered, the sequencer will vector to the specified address of the RUNBIT routine.

The RUNBIT routine would verify operation of the processor circuitry by testing all of its elements. The register stack and all internal Random Access Memory (RAM) can be exhaustively checked. A checksum can be generated for microcode ROM and compared with a previously stored value. All Arithmetic & Logic Unit (ALU) functions can be checked along with the associated flags and status bits. Data can be routed along all points of the internal buses to verify operation of the multiplexing circuitry. RUNBIT could either be run as a subroutine, that is, all status and contents of registers placed on stack before execution and restored after BIT is completed, or it could be a destructive procedure which initializes the processor after completion. For the purposes of the Microdiagnostics BIT description, RUNBIT is assumed to be non-destructive.

Figures 1 and 2 show the Level I and II Block Diagrams for this BIT technique



274_844_57

Figure 1 Level I Block Diagram for Microdiagnostics

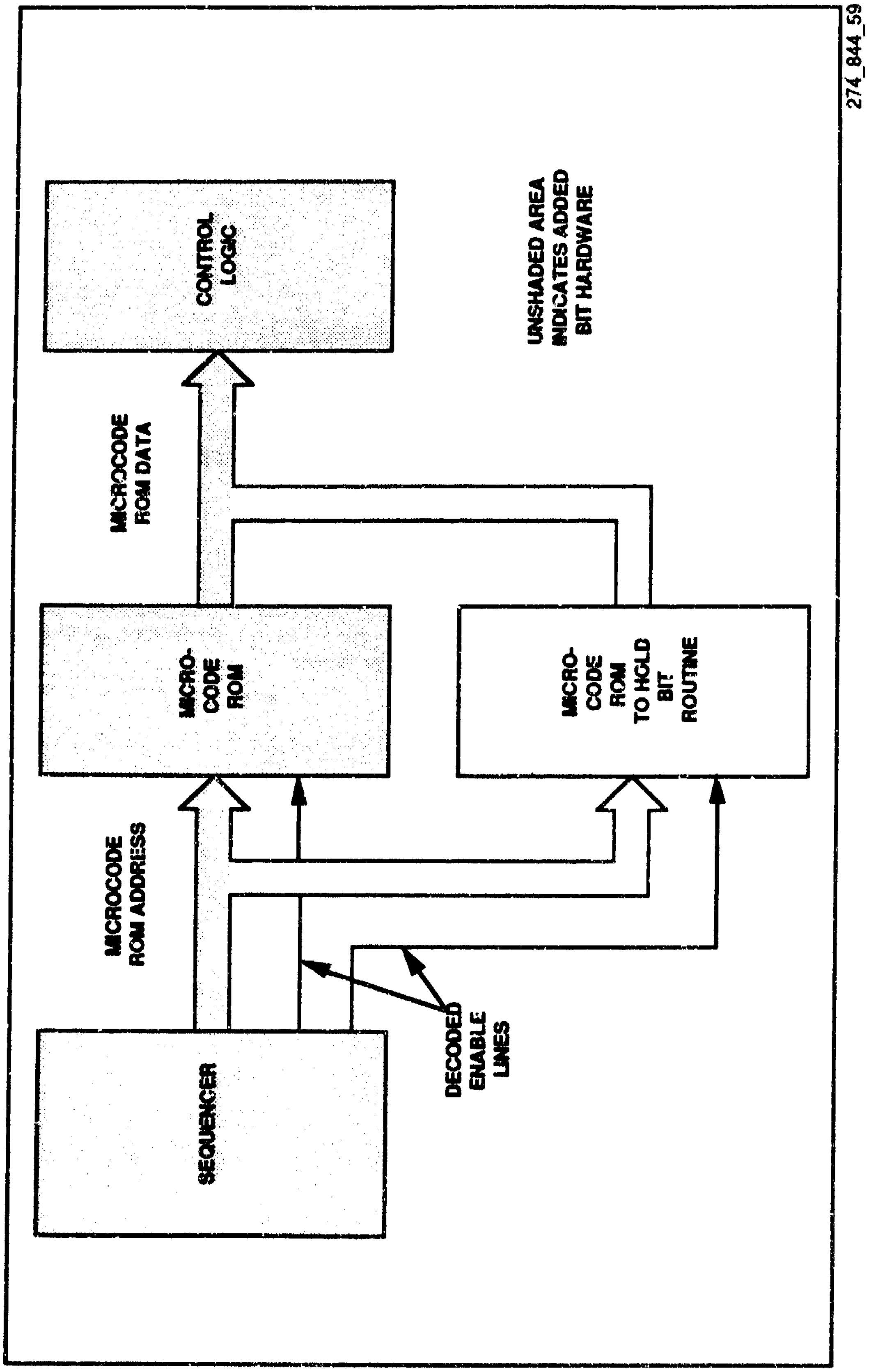


Figure 2 Level II Block Diagram for Microdiagnostics

BIT TECHNIQUE: MICRODIAGNOSTICS

## CATEGORY: FLOW CHART DESCRIPTION

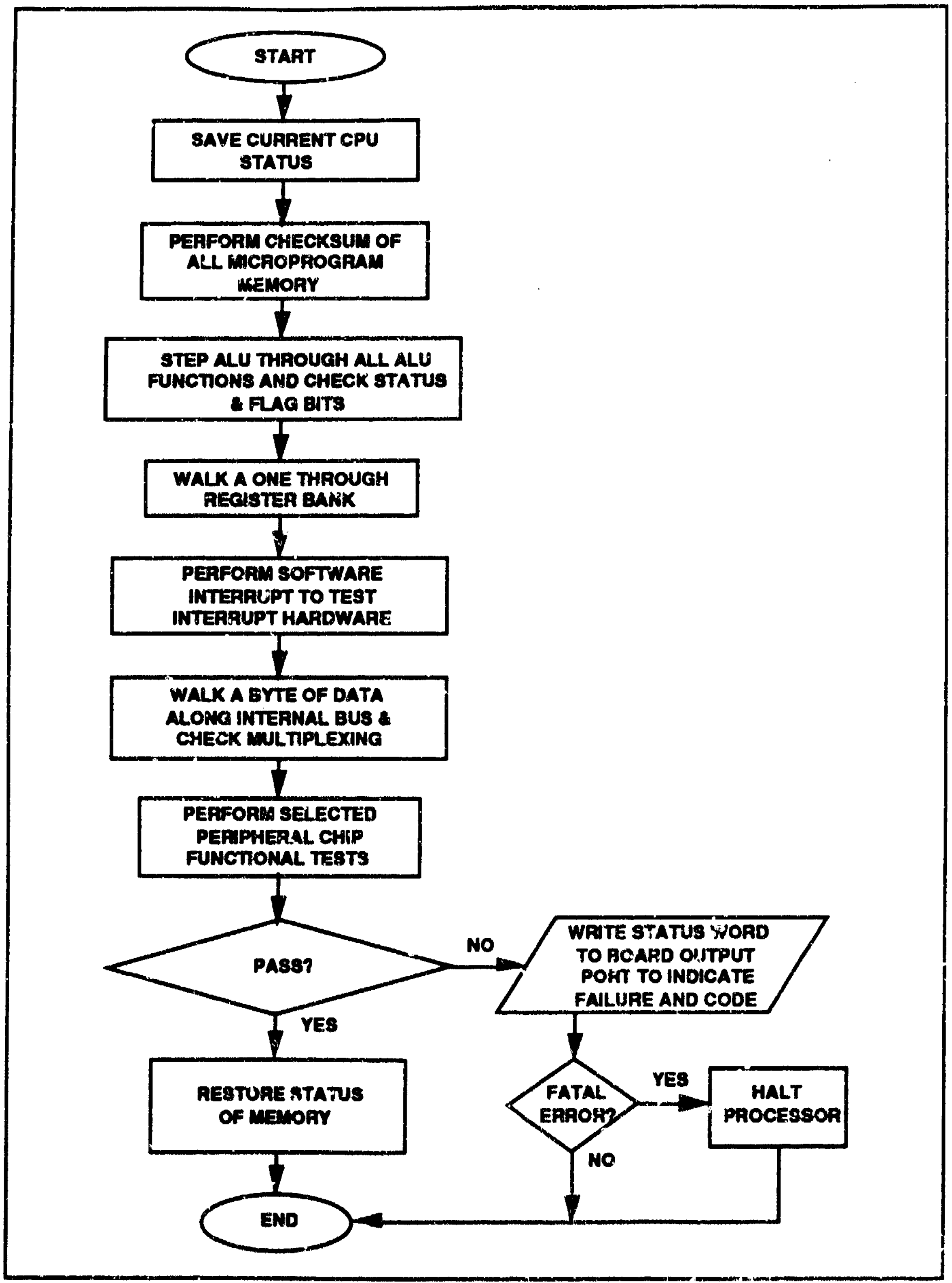
Refer to Figure 3 for a Flow Chart of this BIT technique.

1. Macro instruction "RUNBIT" is called to initiate testing.

   - Save processor status .
   - Perform checksum of all microprogram memory.
   - Step "ALU" through all functions. Check status and flag bits.
   - Walk a one through register banks.
   - Perform software interrupt test.
   - Walk various data patterns along internal bus and check system multiplexing paths.
   - Perform selected functional tests of peripheral devices.

2. If all tests pass, restore processor status.

3. If test fails, write status word to board output port to indicate the failure code and, as an option, halt the processor based on detected fault severity. The result can then be read from the Line Replaceable Module (LRM) by the rest of the system.


START
SAVE CURRENT CPU STATUS
PERFORM CHECKSUM OF ALL MICROPROGRAM MEMORY
STEP ALU THROUGH ALL ALU FUNCTIONS AND CHECK STATUS & FLAG BITS
WALK A ONE THROUGH REGISTER BANK
PERFORM SOFTWARE INTERRUPT TO TEST INTERRUPT HARDWARE
WALK A BYTE OF DATA ALONG INTERNAL BUS & CHECK MULTIPLEXING
PERFORM SELECTED PERIPHERAL CHIP FUNCTIONAL TESTS
PASS?
NO
WRITE STATUS WORD TO BOARD OUTPUT PORT TO INDICATE FAILURE AND CODE
YES
RESTORE STATUS OF MEMORY
FATAL ERROR?
YES
HALT PROCESSOR
NO
END

274_844_58

Figure 3 BIT Sequence Flow Chart for Microdiagnostics

**BIT TECHNIQUE: MICRODIAGNOSTICS**

**CATEGORY: ADVANTAGES**

1. No application software overhead due to microcoded BIT program.

2. BIT will run faster because it will execute at a microinstruction level.

3. Microprocessor BIT can check internal microcomputer circuitry as well as peripheral chip functions.

**CATEGORY: DISADVANTAGES**

1. Possible large hardware requirement due to size of microcode program memory needed to handle BIT test. Example: Large BIT slice configurations with limited memory.

2. Adequate memory may not be available to run all desired tests if the microcode ROM cannot be expanded.

3 Potential major impact to CUT if at least one spare opcode is not available to implement BIT microcode.

## BIT TECHNIQUE: MICRODIAGNOSTICS

## CATEGORY: ATTRIBUTES

1. CONCURRENCY

- Concurrent – BIT must be called infrequently during application software execution.

2. TECHNOLOGY

- Typical application is digital circuitry.
- Applicable to micro-programmable digital processors.
- If additional memory is needed, the same micro-program memory devices are used for BIT program as are used in the base design.

3. CUT MICROPROCESSOR REQUIRED?

- Yes. An existing micro-programmable processor is utilized to test itself and associated peripheral devices.

4. CUT INTERNAL DESIGN REQUIRED?

- No, unless an additional ROM is required.

- BIT microcode is internal to the CUT. Internal hardware design is limited to adding microcode ROM, if required.

5. AREA PENALTY

- None as long as memory space is available for BIT microcode. If memory is not available, microcode ROMs will have to be added proportional to the size of the BIT microcode.

6. WEIGHT PENALTY

- Proportional to the area penalty.

**BIT TECHNIQUE: MICRODIAGNOSTICS**

**CATEGORY: ATTRIBUTES, Contd**

7. POWER PENALTY

- No power penalty if additional ROM is not needed.
- Power penalty is a function of the number and type of needed ROM chips.
- Power penalty may be minimized through use of ROMs with a power-down mode.

8. TIMING PENALTY

- Test Time – Very fast compared to application code BIT implementation. Approximate execution time = (number of BIT micro-instructions) x (average micro-instruction execution time).
- Throughput – Throughput impact is proportional to the frequency that the BIT macro-instruction is called.

9. RELIABILITY IMPACT

- Failure rate increases as the number of additional ROM chips increases.

10. CONCEPTUAL COMPLEXITY

- Straightforward

11. HARDWARE / SOFTWARE / FIRMWARE

- Hardware – Possible additional microcode ROM(s)
- Software -- Addition of BIT macro-instruction call in application code along with a test for failure and a conditional branch statement.
- Firmware -- Test patterns and instructions in microcode

## BIT TECHNIQUE: MICRODIAGNOSTICS

## CATEGORY: ATTRIBUTES, Contd

12. DESIGN COST

- Hardware -- No new design beyond the addition of additional memory device(s) if necessary.
- Software -- Minimal (execute single microcode instruction lines)
- Firmware -- Dependent on complexity of BIT sequence but will typically be less complex than a software implementation.

13. MEMORY REQUIREMENT

- A function of BIT sequence complexity (microcode instructions and test patterns).

14. BIT CIRCUITRY SELF-TESTABLE?

- YES -- BIT microcode ROM will be checksummed along with the CUT microcode.

15. STAND-ALONE (SELF-CONTAINED) BIT?

- YES -- BIT microcode is contained in the processor microcode ROM. The only possible external component is the application code call to the BIT macro instruction.

16. NOTES

- Microcoded BIT technique is contingent on the availability of at least one microcode instruction opcode. If an opcode is not available, then a much more complex change to the CUT processor will be required.

## BIT TECHNIQUE: MICRODIAGNOSTICS

## CATEGORY: DEFAULT DESIGN

Refer to Figure 2 for the Level II Block Diagram for this BIT technique. No default design schematic has been included as part of the microdiagnostics BIT technique. Each application of a microcoded processor design will be unique. The only addition to a design needed to implement Microdiagnostics is one or more microcode ROMs of the same type that are already being used in the design.

**BIT TECHNIQUE: MICRODIAGNOSTICS**

**CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM**

Figure 4 shows the BIT Technique Insertion Diagram for this BIT technique.

1. Connect the BIT microcode ROM address lines in parallel with the existing microcode ROM address lines.

2. Connect the BIT microcode ROM data lines in parallel with the existing microcode ROM data lines.

3. Connect the appropriate chip enable decoder output to the enable pin of the BIT microcode ROM.

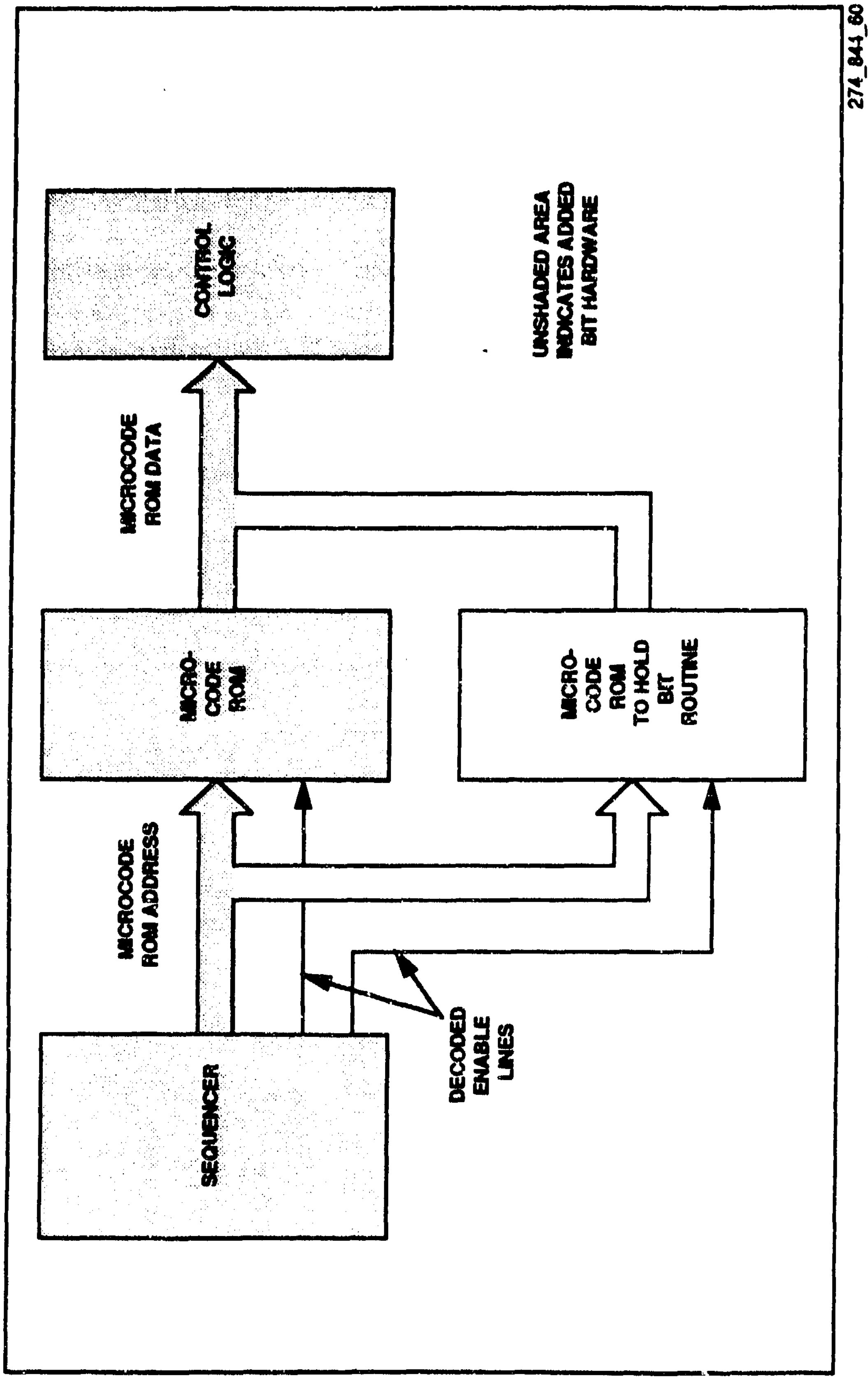


Figure 4 BIT Technique Insertion Diagram

## BIT TECHNIQUE: MICRODIAGNOSTICS

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | #Additional ROMs required for Microdiagnostics BIT | #ROMs |
| $v_2$. | Microdiagnostic BIT execution time | sec |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 27C256-12 | PROM | $v_1$ |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i \quad = \quad \text{ceil}\,(\,u_i\,/\,upp_i\,)$$

**Explanation of symbols used:**

| | | |
|---|---|---|
| $n_i$ | = | Number of components (physical packages) required for i-th part |
| $u_i$ | = | Number of units (CAD symbols) required for i-th part |
| $upp_i$ | = | Number of units/package for i-th part (from Table 4.0) |
| ceil | = | Round up to nearest integer (e.g. 5.3 is rounded to 6) |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |

## BIT TECHNIQUE: MICRODIAGNOSTICS

## CATEGORY: PENALTY EQUATIONS

| | | |
|---|---|---|
| **AREA (sq.in.)** | = | Sum ( $n_i * a_i$ ) + 15% for traces |
| **WEIGHT (gms)** | = | Sum ( $n_i * w_i$ ) + 10% for solder |
| **POWER (mW)** | = | Sum ( $n_i * p_i$ ) |
| **TEST TIME (sec)** | = | v2 |
| **DELAY (ns)** | = | 0 |

**Explanation of Symbols Used.**

| | | |
|---|---|---|
| Sum | = | Sum over parts in BIT Technique Parts Table (i=1 to 1) |
| $n_i$ | = | Number of component packages for i-th part in Parts Table<br>$n_i$ is calculated according to Component Determination Equations |
| $a_i$ | = | Area in sq.in. of i-th part in Parts Table |
| $p_i$ | = | Power dissipation in mW (typical) for i-th part in Parts Table |
| $w_i$ | = | Weight in grams of i-th part in Parts Table |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |

## CATEGORY: BIBLIOGRAPHY

1. T. Sidhar and J.P. Hayes, "Testing bit-sliced Microprocessors" in Proc. 9th Int. Conf. Fault-Tolerant Computing. Madison, WI: IEEE Comp. Soc. June 1979. PP 211-218.

2. The AM2900 Family Databook, Advanced Micro Devices, Sunnyvale, CA., 1976.

3. D.P. Fulghum, "Automatic Self-Test of a Microprocessor System", Proc. Autotestcon 1976, Arlington, Texas, Nov. 1976. PP 47-52 (Abstracts in IEEE Trans, Aerospace and Electronic Systems, Vol. AES-13 No. 2, March 1977).

# ON-BOARD INTEGRATION OF VLSI CHIP BIT (OBIVCB)

## BIT TECHNIQUE DATA PACKAGE

### CATEGORY: SHORT DESCRIPTION / DIAGRAMS

OBIVCB makes use of Built-In-Self-Test (BIST) internal to integrated circuits (ICs) resident on the circuit board. It does this by providing a "Test Processor Node" on the board, which is capable of addressing a variety of chip Built-In-Test approaches. For instance, the Test Processor Node can supply test patterns and collect test data for the ICs which contain internal SCAN/SET circuitry. Additionally it can also test chips which contain BILBO circuitry.

In general, the Test Processor Node will coordinate chip self-testing, allow for parallel testing, and take advantage of the more current techniques of self-test.

Types of BIT supported by OBIVCB:

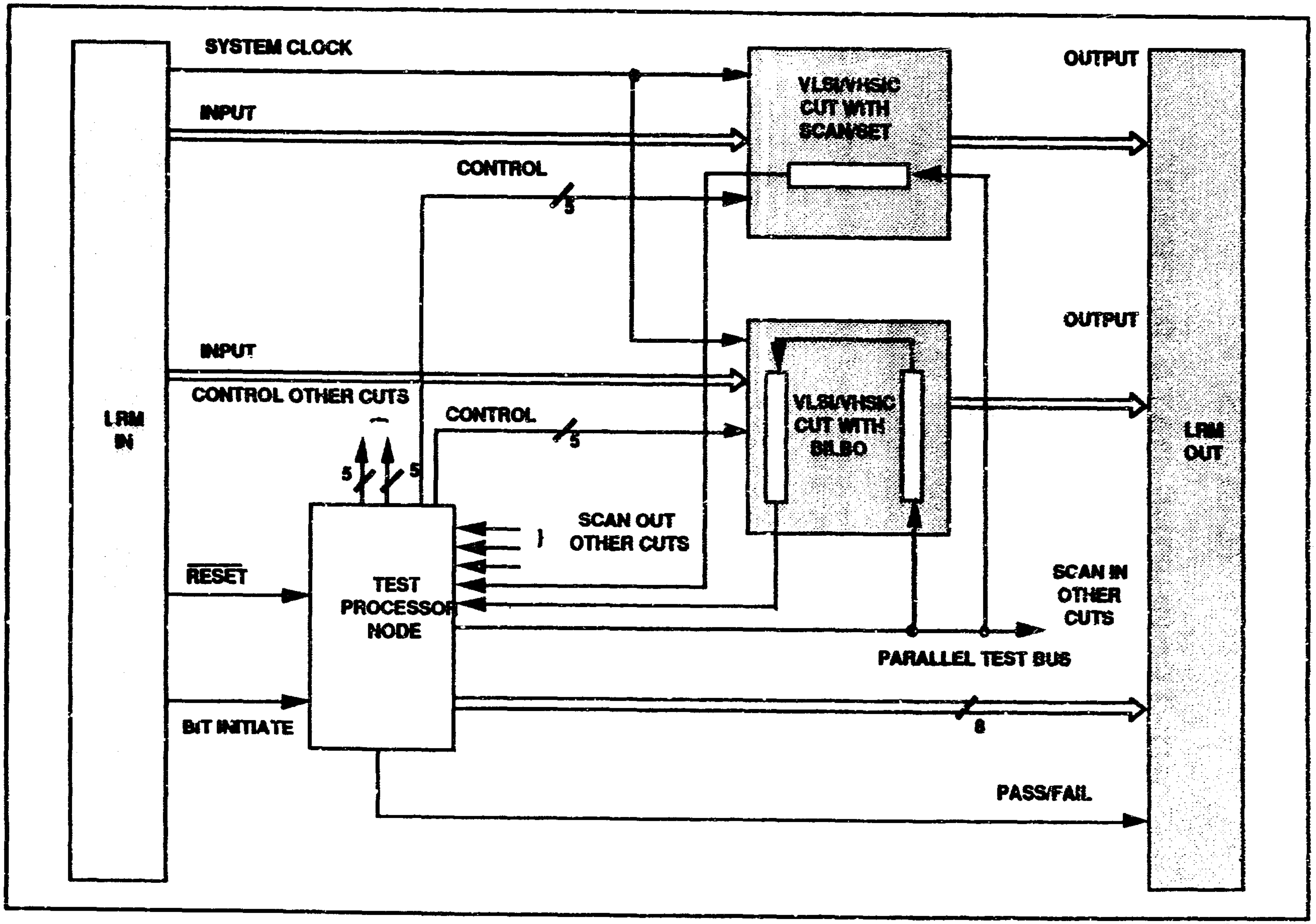
- Scan path techniques
- Internally supported scan, boundary scan, test bus IEEE Std 1149.1 - 1990, i.e. pseudorandom pattern generation and signature analysis is provided on chip.
- Visibility block approach: Built In Logic Block Observer (BILBO), shadow registers, configurable test points, etc.
- Conventional BIST with chip fail or status flags.
- Watchdog timer BIT technique

Figures 1 shows the Level I Block Diagram for this BIT technique. Level II Block Diagrams are provided in Figures 2-4 as follows:

Figure 2 – Level II Block Diagram – Test Processor Node

Figure 3 – Level II Block Diagram – VLSI/VHSIC CUTwith BILBO

Figure 4 – Level II Block Diagram – VLSI/VHSIC CUT with SCAN/SET



274_844_61

Figure 1 Level I Block Diagram for On-Board Integration of VLSI Chip BIT

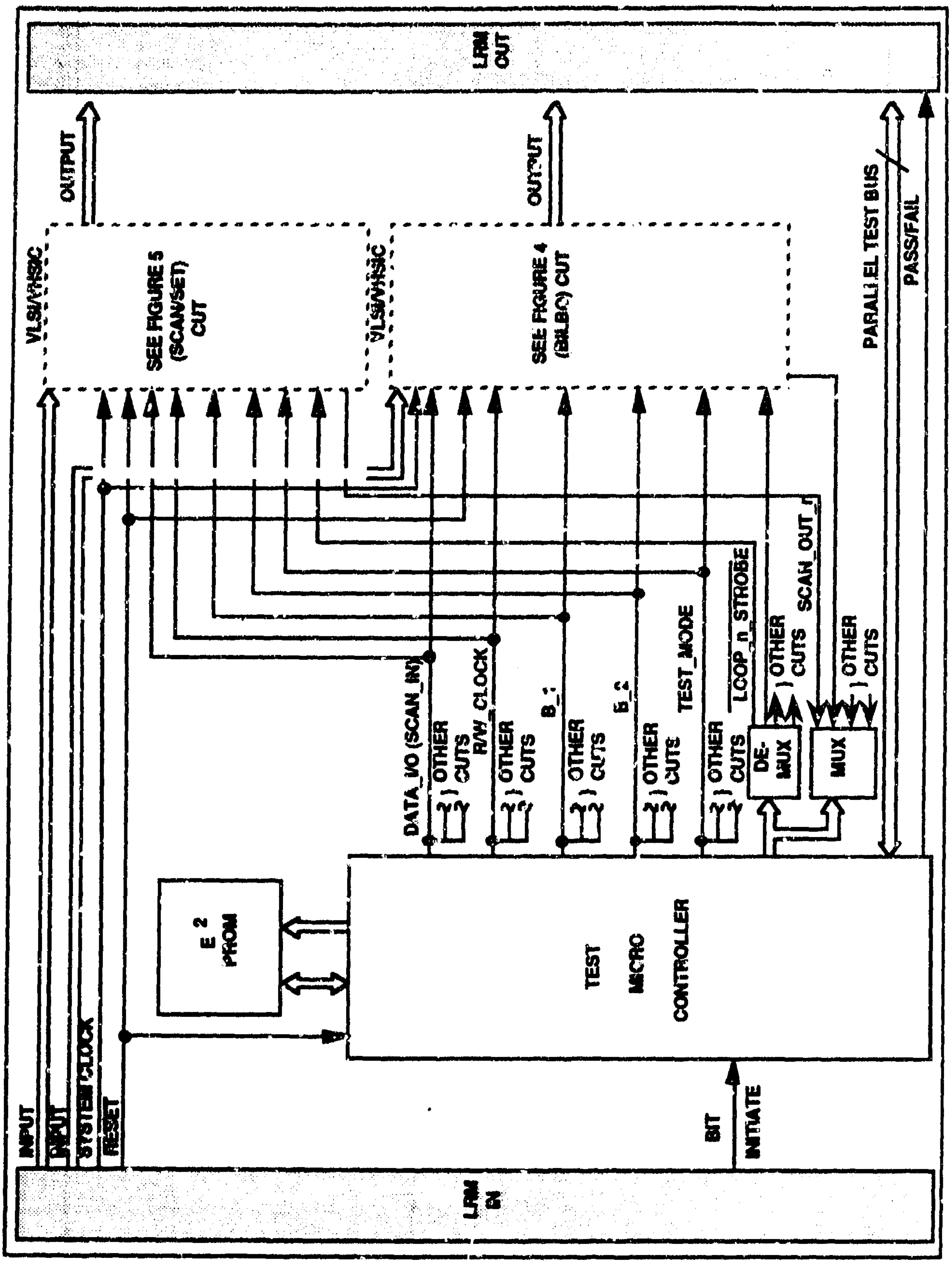
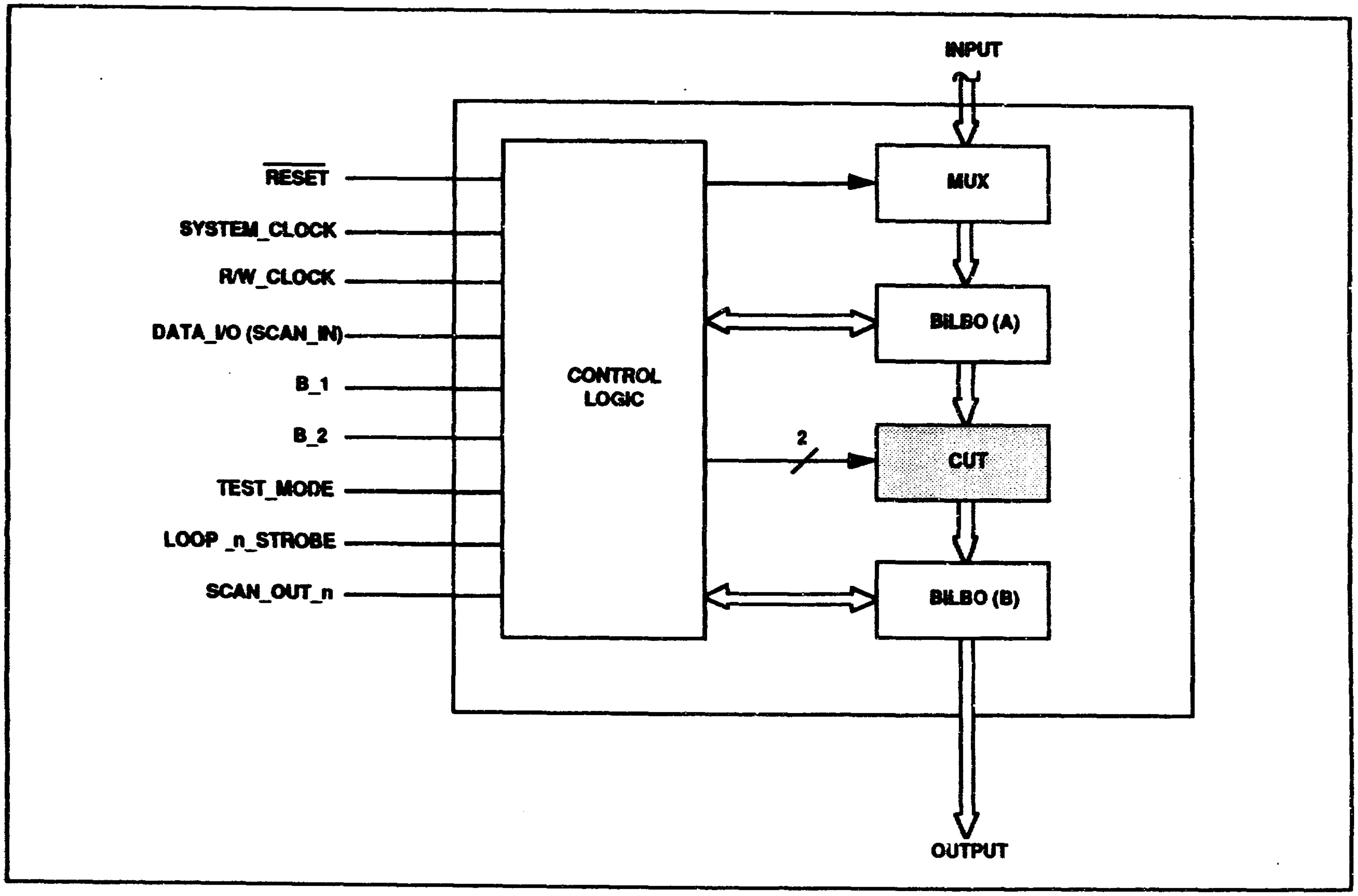


Figure 2 Level II Block Diagram for OBIVCB Test Processor Node



274_844_64

Figure 3 Level II Block Diagram for VLSI/VHSIC CUT with BILBO

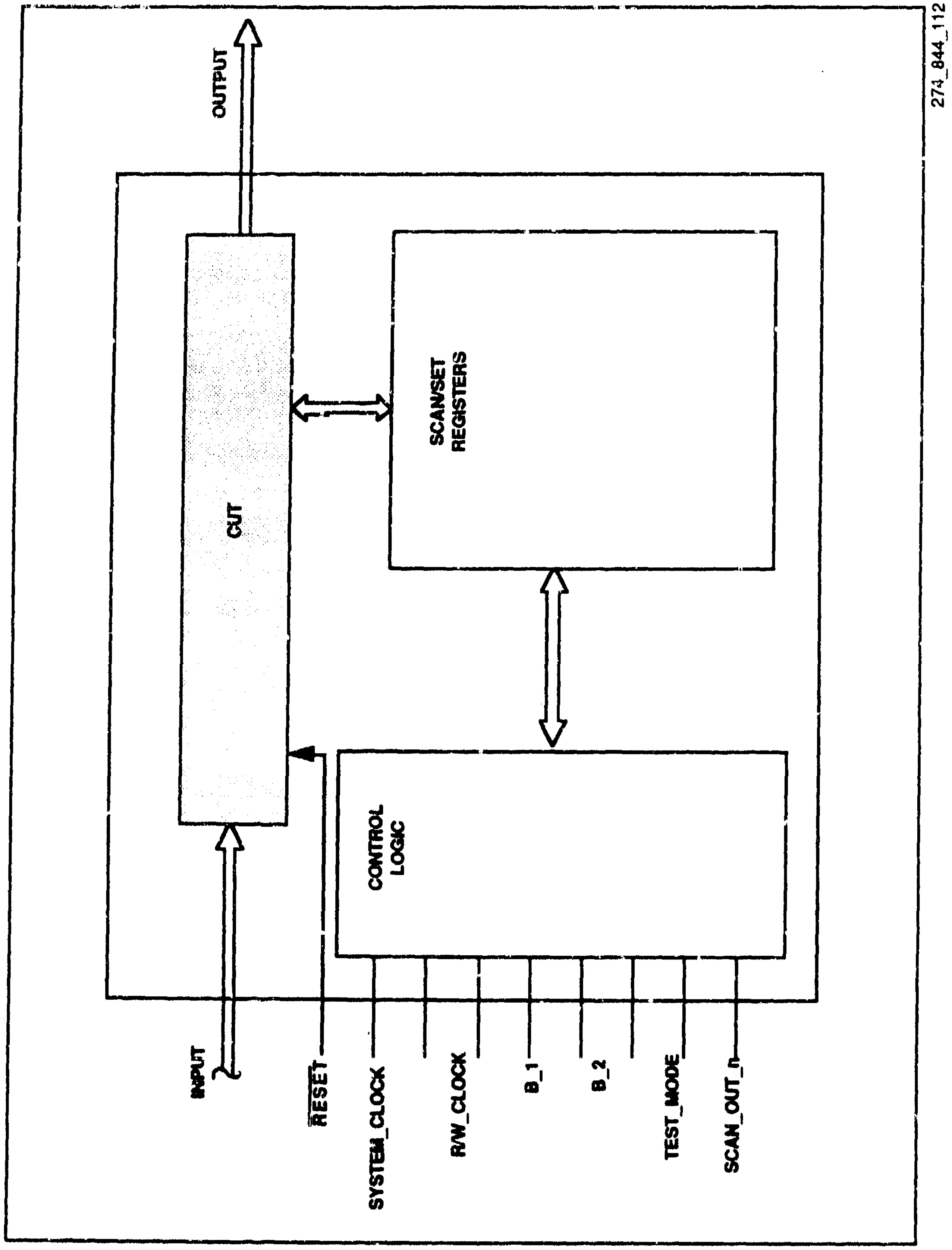


Figure 4 Level II Block Diagram For VLS/VHSIC CUT With Scan/Set

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: FLOW CHART / DESCRIPTION**

Figure 5 shows the Flow Chart for this BIT technique.

1. BIT is initiated externally by the 'BIT INITIATE' pulse. A reset pulse is sent to all chips that can be initialized, as well as interrupting the processor to vector it to a self-test routine. The PASS/FAIL indicator is set to PASS.

2. The Test Processing Node performs a self test. This includes a complete test of all registers, read/write memory, as well as checksum tests on all program memory. The EEPROM is not tested due to limited write cycle life limitations.

3. If the Test Processing Node does not pass the self test, the PASS/FAIL indicator is set to FAIL and the testing process is ended. The CUT registers are left in the transparent (flow-thru) mode. If the Test Processing Node passes the self test, the testing continues.

4. Upon successful completion of Step 3, the processor reads a configuration block in memory which tells it:

   - Which and how many SCAN/SET (S/S) and BILBO registers need to be initialized.
   - Which test loops should be assigned to which registers.
   - Which BILBO registers to place in pattern generation, signature, or scan mode.
   - What patterns to place in SCAN/SET registers
   - What "seeds" to place in BILBO registersp

5. The Test Processor Node selects the first test loop and disables the system clock to all CUTs.

6. If the test loop is implemented with CUTs containing BILBO registers, the following test process continues with Step 7 below. If the test loop is implemented with CUTs containing SCAN/SET registers, the program continues with the SCAN/SET tests in Step 12 below.

<u>BILBO Tests</u>:

7. Seed values are scanned into the BILBO registers.

8. The BILBO registers are placed in the "PRPG/MISR" mode and the CUT is clocked 2^N -1 times, where "N" is the number of BILBO shift register stages.

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: FLOW CHART / DESCRIPTION, Contd**

9. The BILBO registers are placed in the "SCAN" mode and the signature values are scanned from the CUT to the Test Processor Node.

10. The signature values are compared with known good test data stored in EEPROM. If the comparison data does not agree with the known good values, the data is stored in EEPROM and the PASS/FAIL indicator is set to FAIL. If the comparison data agrees with the known good values, the PASS/FAIL indicator remains at PASS.

11. If all test loops have not been tested, the loop sequencer multiplexer is set to the next path and the test program branches control back to Step 6 indicated above. Otherwise, the program branches to Step 16 below.

<u>SCAN/SET Tests</u>:

12. If the CUT SCAN/SET register is to be "SET" to a given pattern, the SCAN/SET register is placed in the "SET" mode, a test pattern is scanned into the CUT, sufficient time is allowed for logic states to settle, and the registers are placed in the "SCAN" mode to retrieve the results. Otherwise, if the register is only scanned, it is loaded with the current results, placed into "SCAN" mode, and the current results are retrieved from the register.

13. The scanned results are compared with known good test data in EEPROM. If the comparison data does not agree with known good values, the data is stored in EEPROM and the PASS/FAIL indicator is set to FAIL. If the comparison data agrees with the known good values, the PASS/FAIL indicator remains at PASS.

14. If all the required test patterns have not been applied, the next test pattern is selected and the test program continues with Step 12 above. Otherwise, the test process continues with Step 15 below.

<u>Check Test Results</u>:

15. If all of the test loops have not been tested, the loop sequencer is set to the next path and the test program branches control back to Step 6 indicated above. Otherwise, the test process continues with Step 16 below.

16. All registers are placed in the transparent (flow thru) mode, all control lines are placed in the normal operations mode, and the system clock is returned to the CUT clock inputs. Testing is complete.

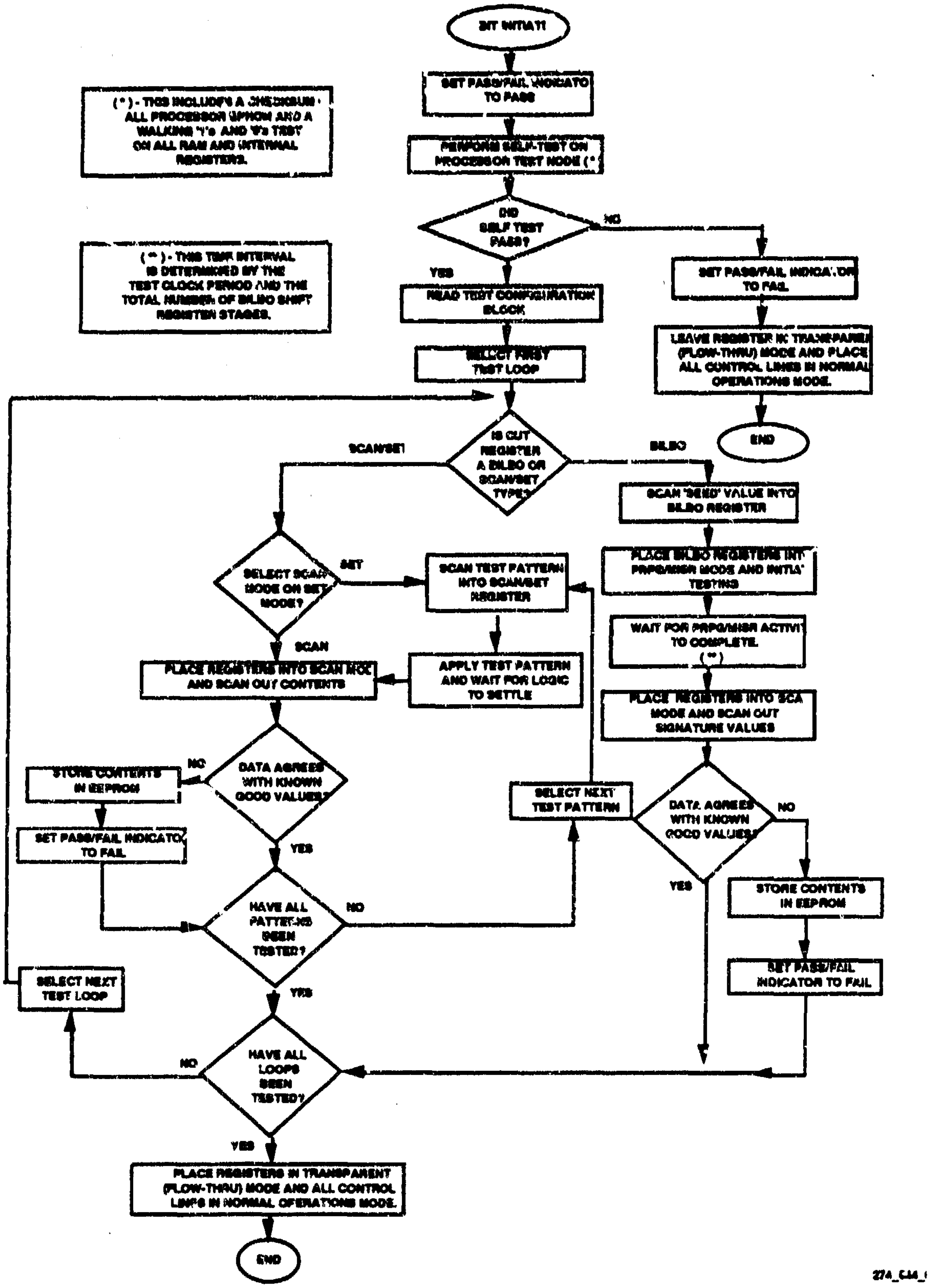

BIT INITIATI
SET PASS/FAIL INDICATO TO PASS
PERFORM SELF-TEST ON PROCESSOR TEST MODE (*
( * ) - THIS INCLUDES A CHECKSUM ALL PROCESSOR EPROM AND A WALKING '1's AND '0's TEST ON ALL RAM AND INTERNAL REGISTERS.
( ** ) - THIS TIME INTERVAL IS DETERMINED BY THE TEST CLOCK PERIOD AND THE TOTAL NUMBER OF BILBO SHIFT REGISTER STAGES.
DID SELF TEST PASS?
NO
YES
SET PASS/FAIL INDICATOR TO FAIL
LEAVE REGISTER IN TRANSPAREN (FLOW-THRU) MODE AND PLACE ALL CONTROL LINES IN NORMAL OPERATIONS MODE.
END
READ TEST CONFIGURATION BLOCK
SELECT FIRST TEST LOOP
IS CUT REGISTER A BILBO OR SCAN/SET TYPE?
SCAN/SET
BILBO
SCAN 'SEED' VALUE INTO BILBO REGISTER
PLACE BILBO REGISTERS INT PRPG/MISR MODE AND INITIA TESTING
WAIT FOR PRPG/MISR ACTIVI TO COMPLETE. (**)
PLACE REGISTERS INTO SCA MODE AND SCAN OUT SIGNATURE VALUES
DATA AGREES WITH KNOWN GOOD VALUES?
NO
YES
STORE CONTENTS IN EEPROM
SET PASS/FAIL INDICATOR TO FAIL
SELECT SCAN MODE OR SET MODE?
SET
SCAN
SCAN TEST PATTERN INTO SCAN/SET REGISTER
APPLY TEST PATTERN AND WAIT FOR LOGIC TO SETTLE
PLACE REGISTERS INTO SCAN MOL AND SCAN OUT CONTENTS
DATA AGREES WITH KNOWN GOOD VALUES?
NO
YES
STORE CONTENTS IN EEPROM
SET PASS/FAIL INDICATO TO FAIL
HAVE ALL PATTERNS BEEN TESTED?
NO
YES
SELECT NEXT TEST PATTERN
SELECT NEXT TEST LOOP
HAVE ALL LOOPS BEEN TESTED?
NO
YES
PLACE REGISTERS IN TRANSPARENT (FLOW-THRU) MODE AND ALL CONTROL LINES IN NORMAL OPERATIONS MODE.
END
274_544_1

Figure 5 BIT Sequence Flow Chart For OBIVCB

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: ADVANTAGES**

1. In defining a Test Processor Node architecture, a standard hardware interface is provided that can be used for widely varying types of chip BIT tests. This standard can later be put into a gate array or standard cell so that implementation can be as simple as designing in a single chip.

2. Because a dedicated processor is used, it is extensible and easily modified by revising firmware.

3. Linear Feedback Shift Register (LFSR) theory, the basis of pattern generation/signature analysis, is well established, documented, and proven. Its high level of fault detection has been the subject of several papers.

4. In tying together various chip BITs with a single Processor Node, a hierarchical test structure is built which is well defined and maintainable. This idea can be extended from the card to the box and system levels.

5. A hardware/software balance is achieved, allowing time costs and chip costs to be worked into project budgets more easily than an approach which is implemented solely in hardware or software.

**CATEGORY: DISADVANTAGES**

1. This approach requires the CUT to be largely self testing; that is, it assumes that BIT is included in most of the chips, especially VLSI and VHSIC devices.

2. A Processor Test Node is required, while not comprising an unreasonable amount of hardware for testing a board of complex logic, may be overkill if the logic is already accessible and not particularly complex. Each application has to be evaluated separately on the basis of need.

3. The test hardware has been optimized for scan path and BILBO type testing. While it can handle other approaches as well, it tends to constrain circuit design, both on the board level as well as the chip level, to this type of structure.

4. It requires control of system clock.

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: ATTRIBUTES**

1. CONCURRENCY

- Test is non-concurrent with operational use.

2. TECHNOLOGY

- Digital

3. CUT MICROPROCESSOR REQUIRED?

- No.

4. CUT INTERNAL DESIGN REQUIRED?

- Yes. CUTs require either BILBO or SCAN/SET registers and supporting control logic.

5. AREA PENALTY

- The area penalty is defined by the base configuration of the Test Processor Node. Actual area for the base configuration is given for Dual-In-Line Packages (DIPs). The BILBO and SCAN/SET area penalties are not provided since they are implemented in the CUT VLSI/VHSIC devices.

6. WEIGHT PENALTY

- Defined by the base configuration of the Test Processor Node. The BILBO and SCAN/SET components are implemented in the CUT VLSI/VHSIC devices.

7. POWER PENALTY

- Defined by the base configuration of the Test Processor Node and increases in proportion to the number of BILBO and SCAN/SET registers used.

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: ATTRIBUTES, Contd**

8. TIMING PENALTY

- Test time – Equal to the sum of the Test Processor Node RAM and EPROM test time plus the test time required to test the BILBO and/or SCAN/SET circuitry. A maximum of four test loops are available that may be any combination of BILBO or SCAN/SET VLSI/VHSIC CUTs.
- Throughput delay
- SCAN/SET – The speed of a CUT containing SCAN/SET registers will be less than one which does not contain SCAN/SET registers due the addition of two port input multiplexing circuitry to the CUT internal latches. The two port feature allows the CUT internal latches to parallel load test pattern data from the SCAN/SET registers.
- BILBO – The speed of a CUT containing BILBO registers will be less than one which does not contain BILBO registers due the addition of an input multiplexer in series with the data lines. The input multiplexer is used to maintain the BILBO(A) input levels at a low state during the PRPG/MISR mode.

9. RELIABILITY IMPACT

- Failure rate increases due to the added BILBO and/or SCAN/SET registers implemented in the CUTs and the Test Processor Node control logic.

10. CONCEPTUAL COMPLEXITY

- Moderate to high complexity

11. HARDWARE/SOFTWARE/FIRMWARE

- Hardware – For the Test Processor Node and added CUT internal BILBO and/or SCAN/SET registers.
- Software – Not applicable.
- Firmware – For the micro-controller EPROM test programming instructions and the external EEPROM storage of BIT failure history data.

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: ATTRIBUTES, Contd**

12. DESIGN COST

- Hardware – Test Processor Node components used are readily available at low cost. The VLSI/VHSIC components require additional registers and logic to implement either the BILBO or SCAN/SET circuitry.
- Software – Not applicable.
- Firmware – The test programming is stored in the micro-controller EPROM.

13. MEMORY REQUIREMENTS

- Micro-controller EPROM memory storage is necessary for the test sequence programming. Additional EEPROM memory storage is necessary for seeds, test vectors, and comparison values of test results.

14. BIT CIRCUITRY SELF-TESTABLE?

- Partially – EPROM and RAM are tested in the micro-controller only. The EEPROM and other supporting BIT circuitry are not self testable.

15. STAND-ALONE (SELF-CONTAINED) BIT?

- No. Supporting BILBO and/or SCAN/SET circuitry are integrated in the VLSI/VHSIC components on board.

16. NOTES

- None.

## BIT TECHNIQUE: OBIVCB

## CATEGORY: DEFAULT DESIGN

Refer to Figures 6-9 for the Default Design schematic. The schematic figures show the following aspects of the default design:

| | |
|---|---|
| Figure 6 | Test Processor Node |
| Figure 7 | SCAN/SET Implementation within VLSI / VHSIC Chip |
| Figure 8 | BILBO Implementation within VLSI / VHSIC Chip |
| Figure 9 | BILBO Register Detail Logic |

### TEST PROCESSOR NODE

The Test Processor Node consists of an Intel 8751 micro-controller, an externally mapped EEPROM data memory device, and test loop select and input data multiplexer logic. The Test Processor Node is capable of controlling up to 4 independent test loops with an extra parallel I/O port that may be used for testing of off-board logic.

The micro-controller contains a 4 K byte EPROM that is used to store the application software for implementation of the BIT algorithms. The 16K (or 2K by 8) EEPROM device is reserved for storage of CUT input test patterns, expected test results, and BIT failure history data. The upper three bits of the EEPROM address lines are used as page control lines. (8 pages of 256 bytes each) and are connected to the upper address port pins P2.0, P2.1, and P2.2. Port P2.7 is dedicated for a PASS/FAIL output and port P3.2 (_INT0) is used for the 'BIT INITIATE' command input. The remaining pins of port 2 and 3 are utilized as data I/O and control lines for the test loops.

When instructed by firmware, the P3.0 (RXD) and P3.1 (TXD) ports are reconfigured from selectable port latches to a serial interface port ("mode 0"). The serial mode allows the Test Processor Node to scan data both into and out of the CUTS via the test loops. The serial port transmits the data in 8 bit increments at a rate of 1/12 of the 8751 clock (12 MHz / 12 = 1 MHz). Port pin P3.0 is used as a bidirectional data line (DATA_I/O) and P3.1 is used for the serial clock (R/W_CLOCK). The DATA_I/O (SCAN_IN) line is used for both the reading and writing of data to the BILBO and/or SCAN/SET registers. The R/W_CLOCK will output a clock cycle for every serial bit written or read from the DATA_I/O (SCAN_IN) line by the Test Processor Node.

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: DEFAULT DESIGN, Contd**

## TEST PROCESSOR NODE, Contd

When the Test Processor Node writes data to the CUT (SCAN_IN), the receiving data line multiplexer (U5) output is tri-stated ("floats") so that data from port P3.0 pin may be serially shifted into the CUT. When the Test Processor Node reads data from the CUT (SCAN_OUT_n), the receiving data line multiplexer (U5) output is enabled so that data from the CUT may be serially shifted into the P3.0 pin.

The P3.1 (TXD) port is also used as a selectable port latch during BILBO MISR/PRPG clocking operations to "manually" perform clocking operations that the serial port "mode 0" function cannot perform. PRPG/MISR requires (2^N)-1 clock cycles where "N" equals the number of BILBO shift register stages. Since the serial port R/W_CLOCK can only transmit in increments of eight (8) clock cycles, the remaining clock cycles are performed by detailed firmware instructions that alternately write '1's and '0's to the port. The BILBO VLSI CUT receives the output port level changes and is clocked during the edge transitions of the resulting R/W_CLOCK output.

As an example, the BILBO CUT detailed in figure 8 contains 8 shift register stages (N=8) and the required clock cycles for (2^N)-1 equals 255. The R/W_CLOCK generates 248 clock cycles as 31 bytes are written out of the serial port. Seven (7) additional clock cycles are required to complete the 255 clock cycles needed. Therefore, the R/W_CLOCK port is reconfigured as a port latch and the remaining 7 clock cycles are generated by writing discrete '1's and '0's to the R/W clock port.

The function of the Test Processor Node during tests of the BILBO and SCAN/SET test loops will be discussed in the following sections.

NOTES:
"BILBO" = Built In Logic Block Observer
"PRPG" = Pseudorandom Pattern Generation
"MISR" = Multiple Input Signature Register.

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: DEFAULT DESIGN, Contd**

BILBO

The normal state for the BILBO control lines within the VLSI CUT are B1 and B2 =1 The normal state for the Test Processor Node prior to entering a test of a BILBO loop is:

| | | | |
|---|---|---|---|
| a. | _LOOP_n_STROBE | = | logic "1" |
| b. | TEST_MODE | = | logic "0" |
| c. | B_1 and B_2 | = | logic "0" |

The sequence of operations for the BILBO loop are as follows:

1. Scan a test input pattern "seed" into the BILBO shift registers.

   Procedure:

   a. Select a test loop by writing the test loop address to P3.4 and P3.5 of the 8751.

   b. Disable the system clock to the CUT by setting TEST_MODE to logic "1" (P2.3 of the 8751). This action disables the _RESET and SYSTEM_CLOCK lines to the CUT and are replaced by the _LOOP_n_STROBE and R/W_CLOCK lines of the Test Processor Node respectively.

   c. Disable the serial input multiplexer by setting P2.4 of the 8751 to a logic "1" state. This action tri-states ("floats") the output of the serial input multiplexer so that the Test Processor Node can output data from the DATA_I/O (SCAN_IN) line to the CUT.

   d. Configure the 8751 serial port into the write mode.

   e. Generate a _LOOP_n_STROBE by outputting a positive going pulse to the G1 pin of the ALS138 decoder (P3.3 of the 8751). This action clocks the B_1 and B_2 values of the Test Processor Node to the respective B1 and B2 inputs of the CUT BILBO registers (shift mode) and momentarily resets the CUT logic.

   f. Write the appropriate number of bytes out of the 8751 serial port. This action serially shifts the "seed" information from the Test Processor Node to the CUT BILBO registers. Note that the R/W_CLOCK will output a clock cycle for every serial bit written.

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: DEFAULT DESIGN, Contd**

BILBO, Contd

2. Set the BILBO register to the "PRPG/MISR" mode and clock the registers (2^N) -1 times (N is the number of shift registers in the BILBO (A) register).

   Procedure:

   a. Set B_1 control line to logic "1" (P2.5 of 8751).

   b. Generate a _LOOP_n_STROBE by outputting a positive going pulse to the G1 pin of the ALS138 decoder (P3.3 of the 8751). This action clocks the B_1 and B_2 values of the Test Processor Node to the respective B1 and B2 inputs of the CUT BILBO registers (PRPG/MISR mode) and momentarily resets the CUT logic.

   c. The registers are clocked (2^N) - 1 times by writing the appropriate number of bytes out of the 8751 serial port and filling in the remaining clock cycles with the R/W_CLOCK port reconfigured as a selectable port latch. For more information regarding operation of the serial port, please refer to the Default Design Description of the Test Processor Node.

3) SCAN the signature data out of the BILBO registers.

   Procedure:

   a. Set B_1 control line to logic "0" (P2.5 of 8751).

   b. Generate a _LOOP_n_STROBE by outputting a positive going pulse to the G1 pin of the ALS138 decoder (P3.3 of the 8751). This action clocks the B_1 and B_2 values of the Test Processor Node to the respective B1 and B2 inputs of the CUT BILBO registers (shift mode) and momentarily resets the CUT logic.

   c. Configure the 8751 serial port into the read mode.

   d. Enable the multiplexer tri-state output by setting P2.4 of the 8751 to a "0" state. This action connects the "SCAN_OUT_n" to the serial input DATA_I/O (SCAN_IN) of the Test Processor Node.

   e. Perform the appropriate number of read operations (8 bits per 8751 read instruction.). This action serially shifts the "signature" information from the CUT BILBO registers to the Test Processor Node. Note that the R/W_CLOCK will output a clock cycle for every serial bit written.

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: DEFAULT DESIGN, Contd**

BILBO, Contd

4. Set BILBO register to transparent (flow-thru) mode.

   Procedure:

   a. Set B_1 control line to logic "1". (P2.5 of the 8751).

   b. Set B_2 control line to logic "1". (P2.5 of the 8751).

   c. Generate a _LOOP_n_STROBE by outputting a positive going pulse to the G1 pin of the ALS138 decoder (P3.3 of the 8751). This action clocks the B_1 and B_2 values of the Test Processor Node to the respective B1 and B2 inputs of the CUT BILBO registers and momentarily resets the CUT logic.

   d. Set B_1 and B_2 control lines to logic "0".

   e. Enable the system clock by setting TEST_MODE to logic "0" (P2.3 of the 8751).

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: DEFAULT DESIGN, Contd**

SCAN/SET

The normal state for the Test Processor Node prior to entering a test of a SCAN/SET loop is:

| | | | |
|---|---|---|---|
| a. | _LOOP_n_STROBE | = | logic "1" |
| b. | TEST_MODE | = | logic "0" |
| c. | B_1 and B_2 | = | logic "0" |

The allowable operations for the SCAN / SET loop are as follows:

1. Scan a test input pattern into the SCAN/SET shift register.

   Procedure:

   a. Disable the serial input multiplexer by setting P2.4 of the 8751 to a logic "1" state. This action tri-states ("floats") the output of the serial input multiplexer so that the Test Processor Node can output data from the DATA_I/O (SCAN_IN) line to the CUT.

   b. Configure the 8751 serial port into the write mode.

   c. Write the appropriate number of bytes out of the 8751 serial port (Note that all SCAN/SET loops will be loaded at the same time). This action serially shifts the test pattern from the Test Processor Node to the CUT SCAN/SET registers. Note that the R/W_CLOCK will output a clock cycle for every serial bit written.

2. Scan CUT test data out of the SCAN/SET shift register.

   Procedure:

   a. Select a test loop via the ALS253 multiplexer by writing the test loop address to P3.4 and P3.5 of the 8751. This action selects the output of the CUT "SCAN_OUT_n" to be tested.

   b. Configure the 8751 serial port into the read mode.

   c. Enable the multiplexer tri-state output by setting P2.4 of the 8751 to a "0" state. This action connects the "SCAN_OUT_n" to the serial input DATA_I/O (SCAN_IN) of the Test Processor Node.

   d. Perform the appropriate number of read operations. This action serially shifts the test data from the CUT SCAN/SET registers to the Test Processor Node. Note that the R/W_CLOCK will output a clock cycle for every serial bit written.

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: DEFAULT DESIGN, Contd**

SCAN/SET, Contd

3. Parallel load the shift register with test data from the CUT logic.

   Procedure:

   a. If desired, disable the system clock to the CUT by setting TEST_MODE to logic "1" (P2.3 of 8751).

   b. Select a test loop via the ALS253 multiplexer by writing the test loop address to P3.4 and P3.5 of the 8751. This action selects the output of the CUT "SCAN_OUT_n" to be tested.

   c. Set B_1, B_2 control lines to logic 0, 1 respectively (P2.5 , P2.6 of 8751).

   d. Generate a _LOOP_n_STROBE by outputting a positive going pulse to the G1 pin of the ALS138 decoder (P3.3 of 8751). This action loads the contents of the CUT latches into the SCAN/SET shift registers.

4. Parallel load the CUT latches with data from the shift register.

   Procedure:

   a. If desired, disable the system clock to the CUT by setting TEST_MODE to logic "1" (P2.3 of 8751).

   b. Select a test loop via the ALS253 multiplexer by writing the test loop address to P3.4 and P3.5 of the 8751. This action selects the output of the CUT "SCAN_OUT_n" to be tested.

   c. Set B_1, B_2 control line to logic 1, 0 respectively (P2.5 , P2.6 of 8751).

   d. Generate a _LOOP_n_STROBE by outputting a positive going pulse to the G1 pin of the ALS138 decoder (P3.3 of 8751). This action loads the contents of the SCAN/SET shift register into the CUT latches.

All CUT logic using SCAN/SET loops will be restored to normal operation when TEST_MODE control line is returned to logic "0" (P2.3 of 8751).

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: DEFAULT DESIGN, Contd**

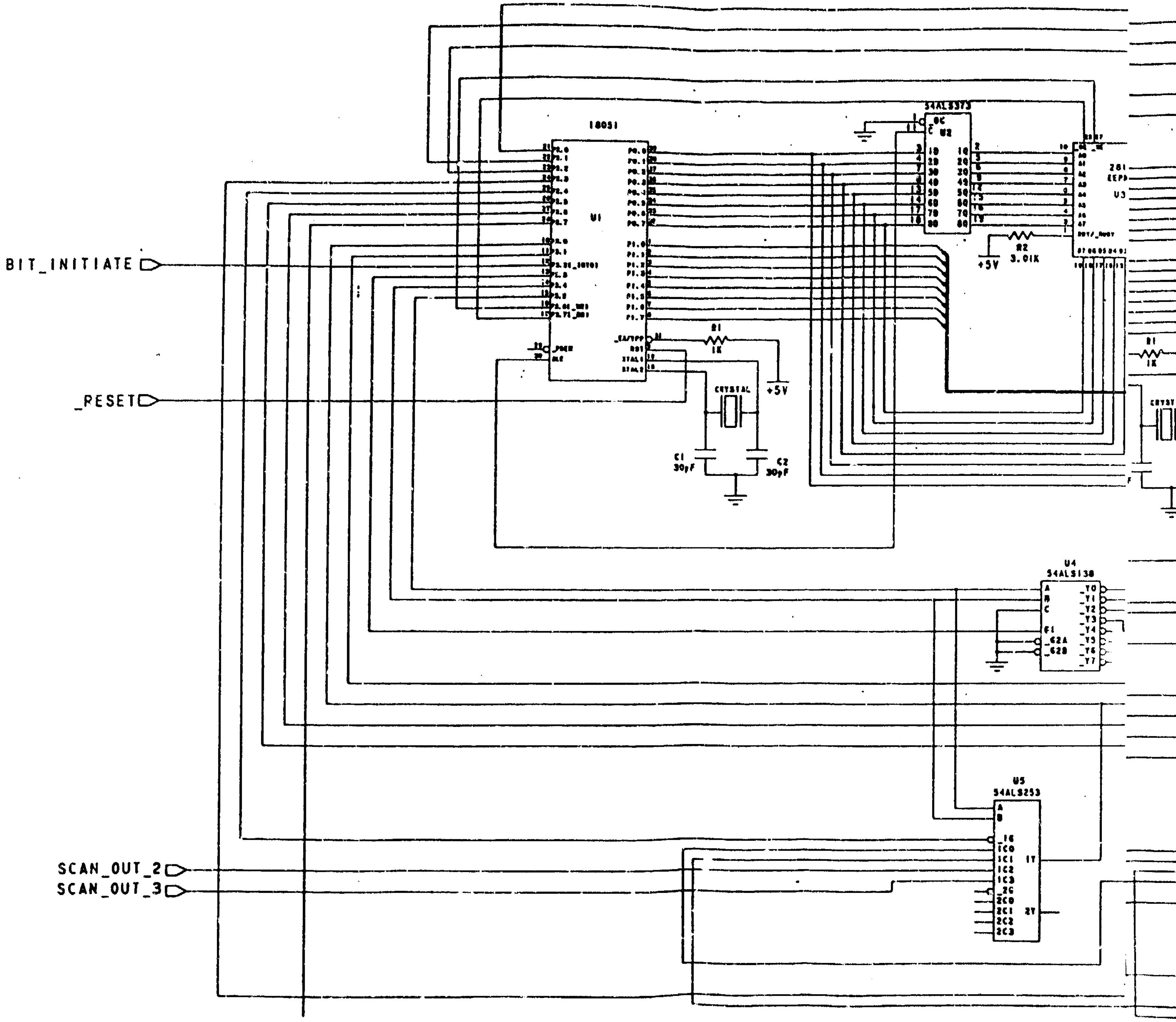
LIMITATIONS

1. All VLSI/VHSIC chips will be disabled from normal operation when testing of any loop is in progress. The system clock is disabled and replaced with the microprocessor read/write clock.

2. The parallel test bus has limited capability due to a lack of dedicated address and control data lines.

3. All Test Processor Node read/write clock cycles used to clock the CUT during testing are in 8 clock cycle increments.

4. Current implementation lacks a dedicated communication port to report detailed BIT results and failure histories.


I8051
U1
54ALS373
U2
28I
EEPROM
U3
R2
3.01K
+5V
R1
1K
CRYSTAL
C1
30pF
C2
30pF
U4
54ALS138
U5
54ALS253
BIT_INITIATE
_RESET
SCAN_OUT_2
SCAN_OUT_3

Figure

(1)

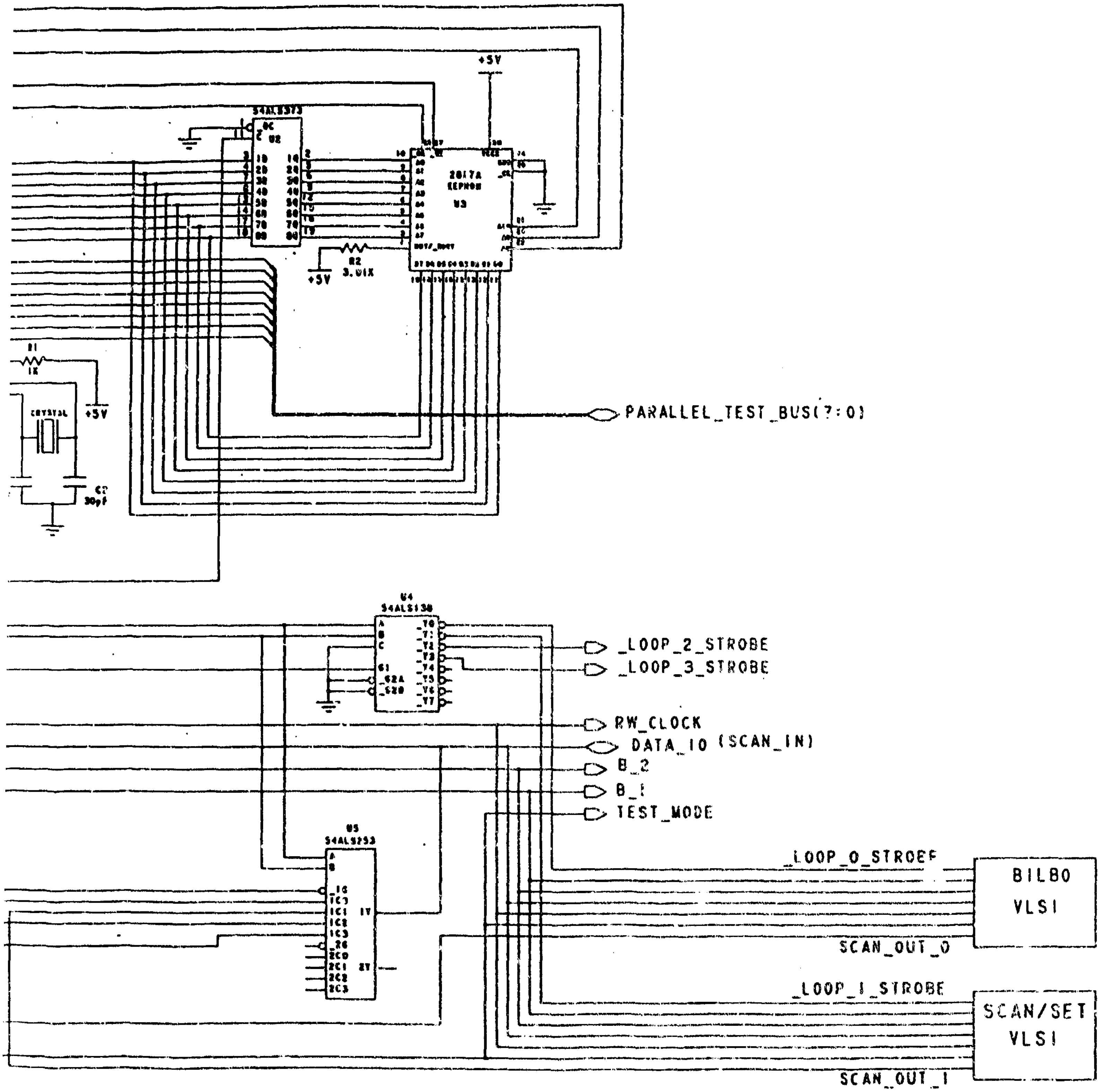


Figure 6 Test Processor Node Default Design for OBIVCB

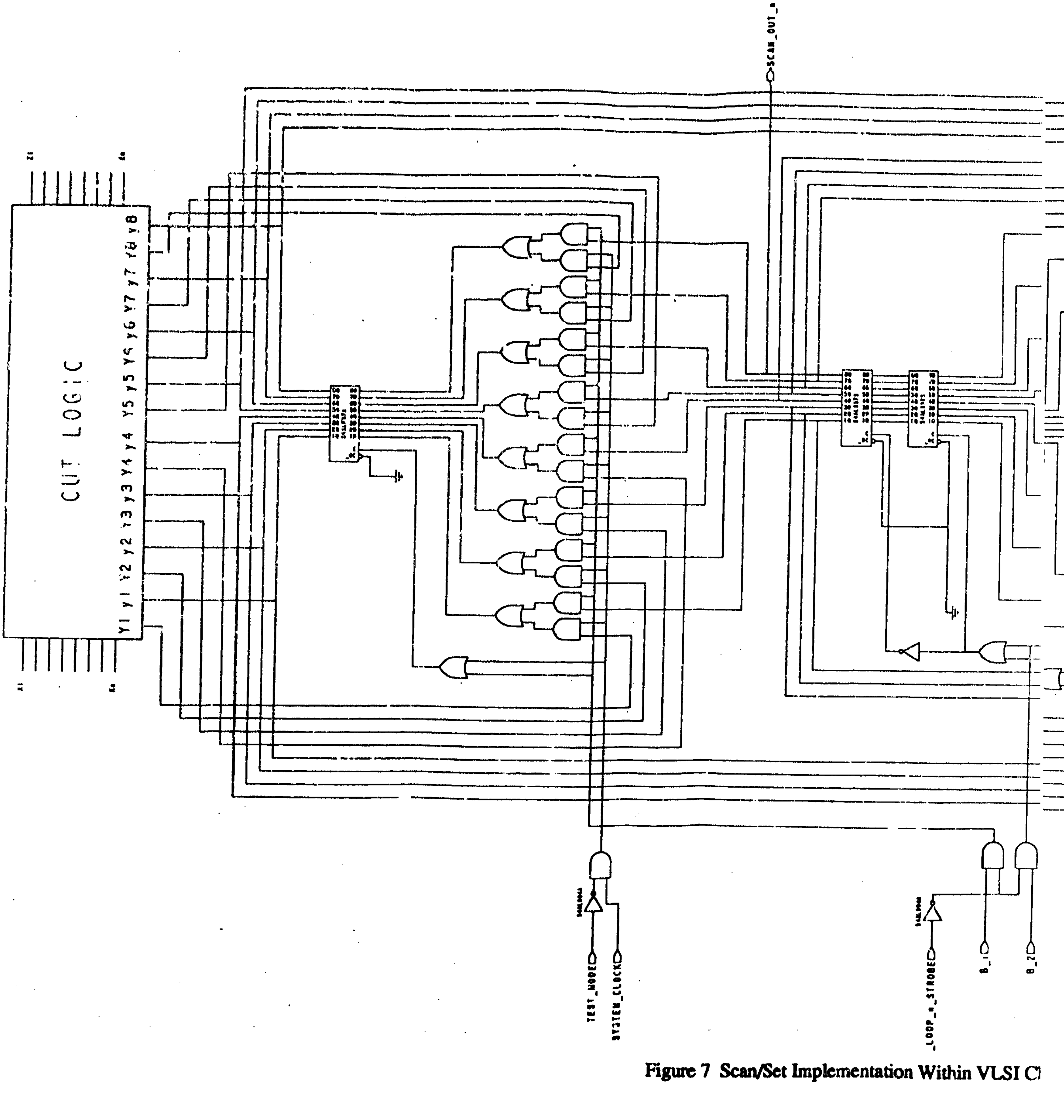


Figure 7 Scan/Set Implementation Within VLSI C

1

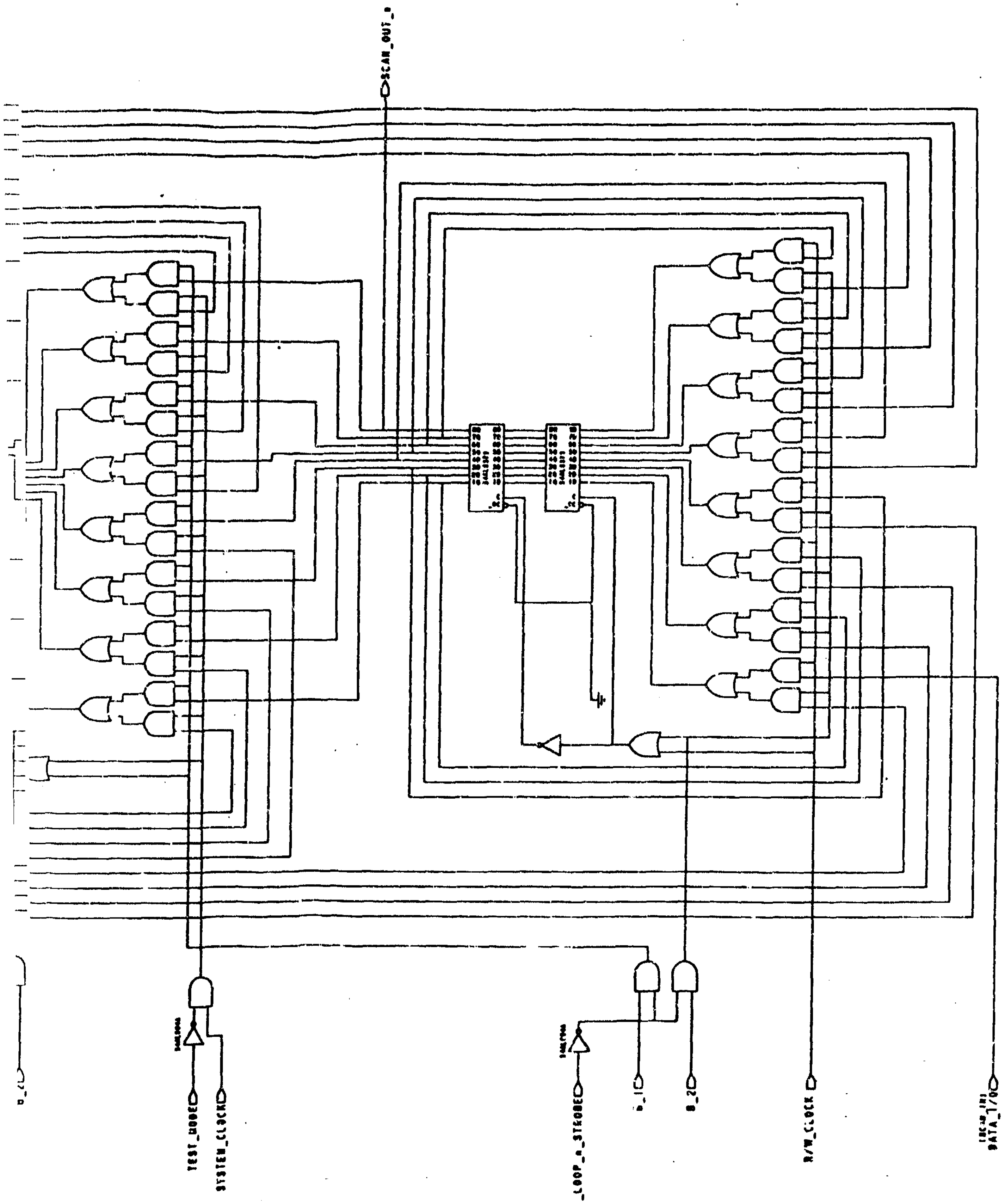


Figure 7 Scan/Set Implementation Within VLSI Chip – Default Design

2

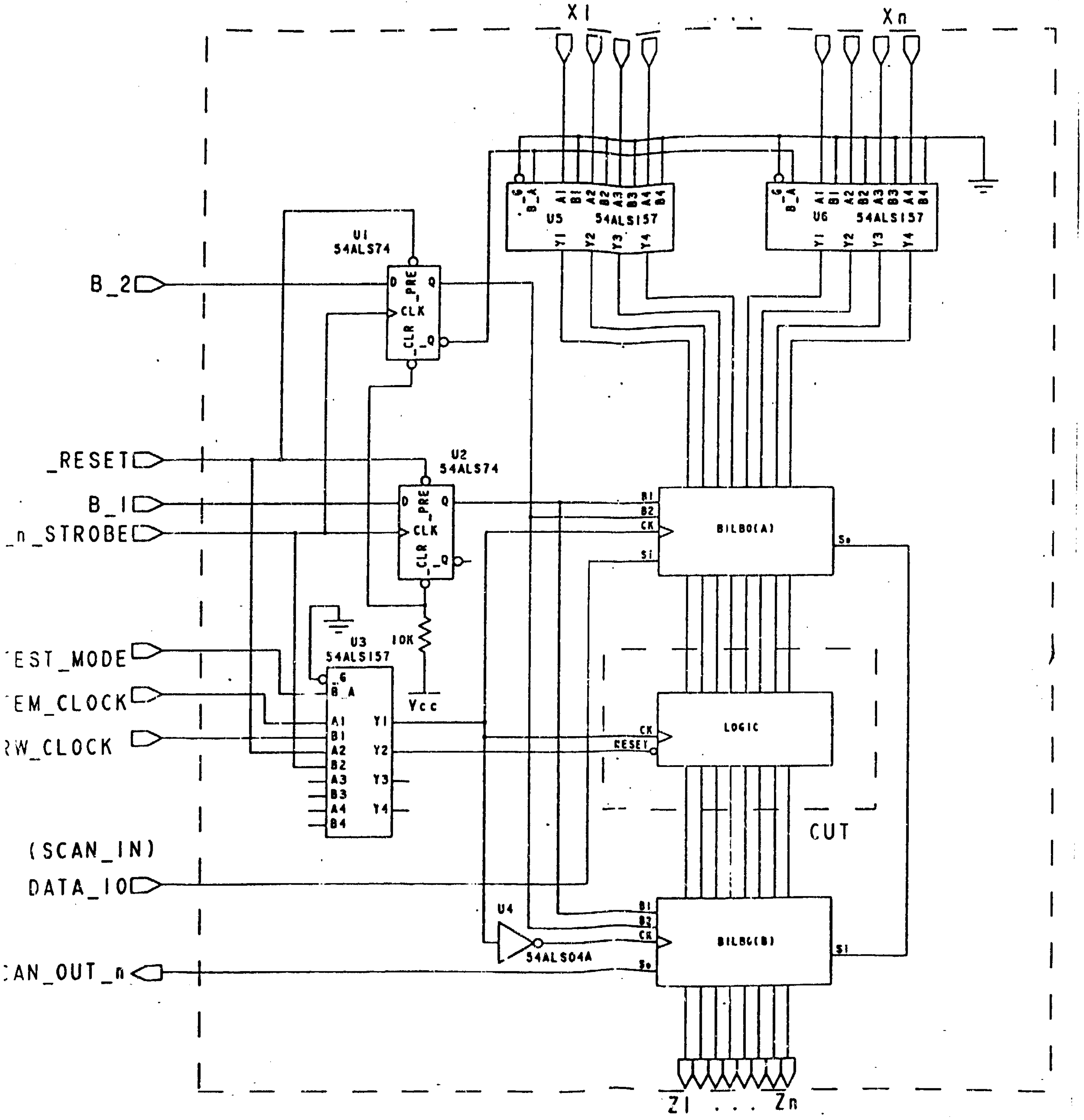


Figure 8 BILBO Implementation Within VLSI/VHSIC Chip – Default Design

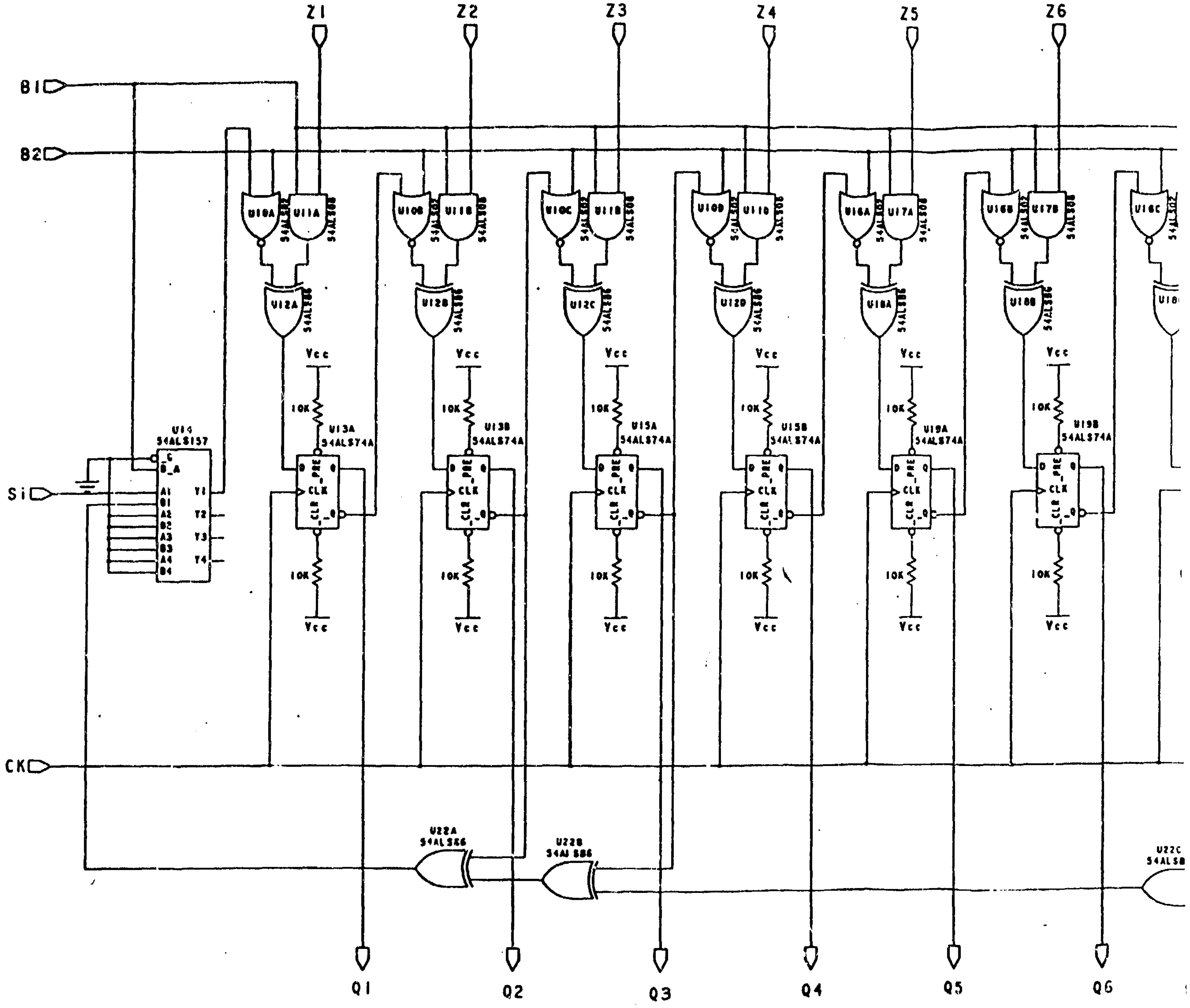
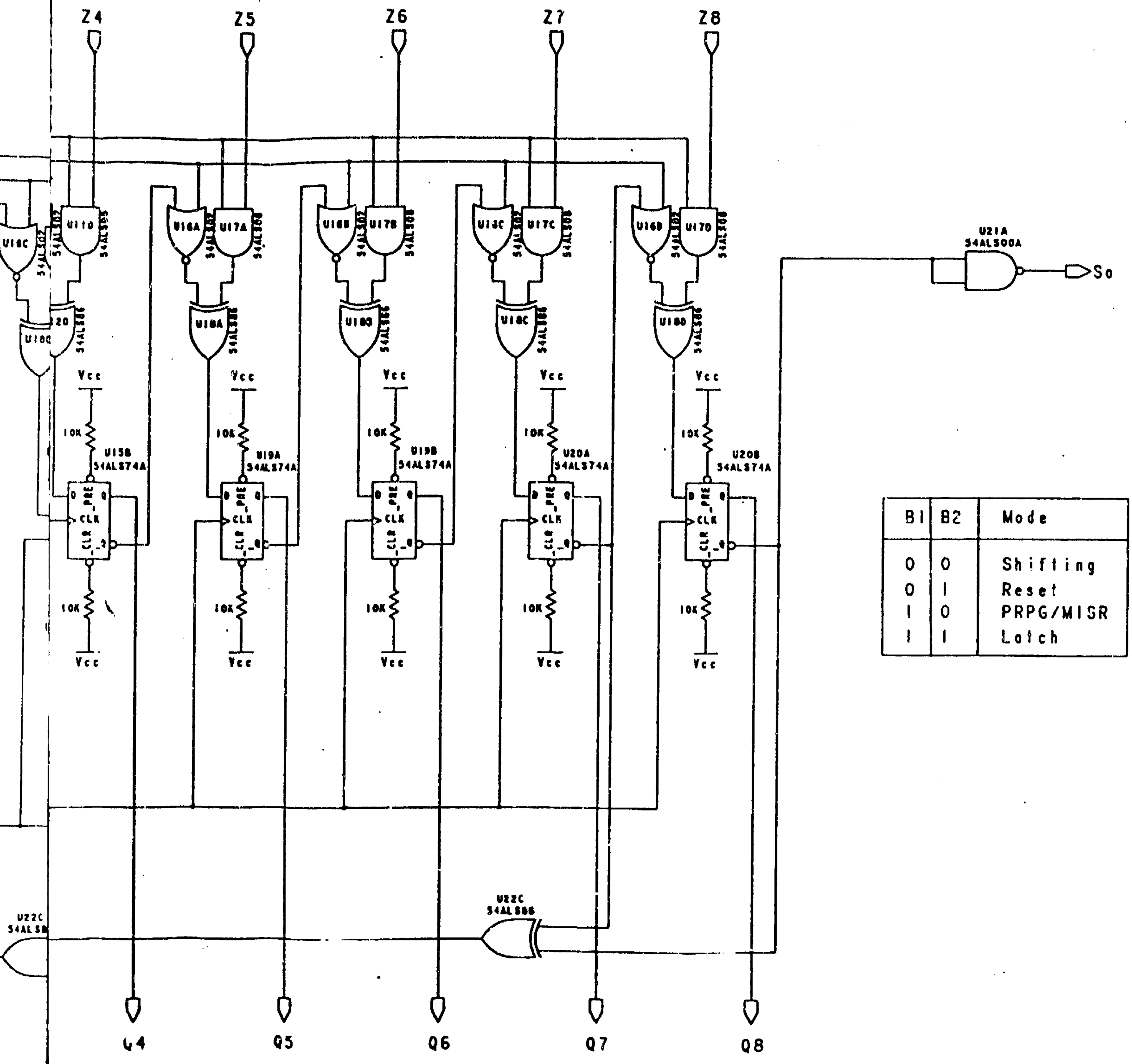

Z1
Z2
Z3
Z4
Z5
Z6
B1
B2
U10A 54ALS02
U11A 54ALS08
U10B 54ALS02
U11B 54ALS08
U10C 54ALS02
U11C 54ALS08
U10D 54ALS02
U11D 54ALS08
U16A 54ALS02
U17A 54ALS08
U16B 54ALS02
U17B 54ALS08
U16C 54ALS02
U12A 54ALS86
U12B 54ALS86
U12C 54ALS86
U12D 54ALS86
U18A 54ALS86
U18B 54ALS86
Vcc
10K
U13A 54ALS74A
U13B 54ALS74A
U15A 54ALS74A
U15B 54ALS74A
U19A 54ALS74A
U19B 54ALS74A
D
PRE
Q
CLK
CLR
Q
U14 54ALS157
G
B_A
A1
B1
A2
B2
A3
B3
A4
B4
Y1
Y2
Y3
Y4
Si
CK
U22A 54ALS86
U22B 54ALS86
U22C 54ALS8
Q1
Q2
Q3
Q4
Q5
Q6

Figure 9 B

0



| B1 | B2 | Mode |
|---|---|---|
| 0 | 0 | Shifting |
| 0 | 1 | Reset |
| 1 | 0 | PRPG/MISR |
| 1 | 1 | Latch |

Figure 9 BILBO Register Detail Logic – Default Design

2

## BIT TECHNIQUE: OBIVCB

## CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM

Figure 10 shows the BIT Technique Insertion Diagram for this BIT technique.

BTID 1 (Test Processor Node)

1. Connect the BIT_INITIATE line to the appropriate LRM interface pin.
2. Connect the _RESET signal line to the master reset signal on the LRM.
3. Connect PASS_FAIL output signal to the appropriate LRM interface pin.
4. Connect the PARALLEL_TEST_BUS signal lines (8 total) to unused LRM interface pins. The signal originating at U1 pin 8 (P1.7) is the Most Significant Bit (MSB) of the bus. Subsequent connection to these lines (off-board) may be made as a user option.

   If any of the 4 test loops of the TEST PROCESSOR NODE are to be used off board then perform the following additional connection steps.

5. Connect the R/W_CLOCK, DATA_I/O (SCAN_IN), PASS/FAIL, B_1, B_2, and TEST_MODE signal lines to unused LRM interface pins.
6. For each test loop ("n" = loop number) to be connected off-board:
   a. Connect the appropriate _LOOP_n_STROBE signal lines from U4 to unused LRM interface pins.
   b. Connect the appropriate SCAN_OUT_n signal lines from U5 to unused LRM interface pins.

BTID 2 (BILBO)

1. Connect the B_1, B_2, TEST_MODE, R/W_CLOCK, and DATA_I/O (SCAN_IN) signal lines, through unused VLSI device interface pins, to the appropriate signal interface pins of the Test Processor Node (BTID 1).
2. Connect the _RESET and SYSTEM_CLOCK signal lines, through VLSI device pins, to the LRM master reset and system clock signals.
3. Connect the appropriate _LOOP_n_STROBE signal line, through an unused VLSI device interface pin, to the appropriate _LOOP_n_STROBE signal line from the Test Processor Node (BTID 1).
4. Connect the appropriate SCAN_OUT_n signal line, through an unused VLSI device interface pin to the appropriate SCAN_OUT_n signal line on the Test Processor Node (BTID 1).

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM, Contd**

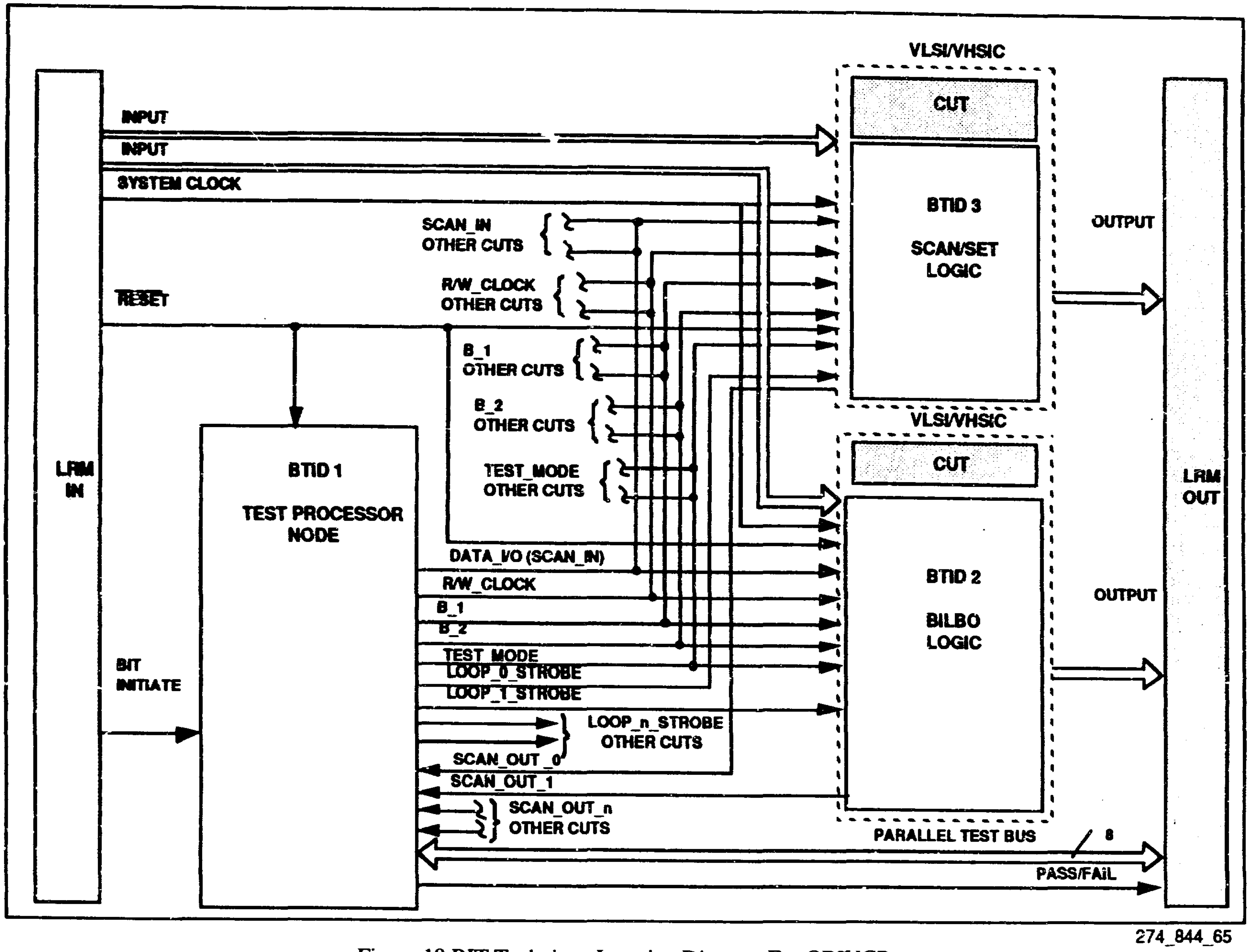
BTID 3 (SCAN / SET)

1. Connect the B_1, B_2, TEST_MODE, R/W_CLOCK, and DATA_I/O (SCAN_IN) signal lines, through unused VLSI device interface pins, to the appropriate signal interface pins of the Test Processor Node (BTID 1).

2. Connect the SYSTEM_CLOCK signal line, through a VLSI device interface pin, to the LRM system clock signal.

3. Connect the appropriate _LOOP_n_STROBE signal line, through an unused VLSI device interface pin, to the appropriate _LOOP_n_STROBE signal line from the Test Processor Node (BTID 1).

4. Connect the appropriate SCAN_OUT_n signal line, through an unused VLSI device interface pin, to the appropriate SCAN_OUT_n signal line on the Test Processor Node (BTID 1).



Figrue 10 BIT Technique Insertion Diagram For OBIVCB

## BIT TECHNIQUE: OBIVCB

## CATEGORY: VARIABLE DEFINITIONS

None required.

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 54ALS138 | DECODER/DEMUX | 1 |
| 2 | 54ALS253 | SELECTOR/MUX | 1 |
| 3 | 54ALS373 | LATCH | 1 |
| 4 | I8751H | MICROCONTROLLER | 1 |
| 5 | M55310/18-12 | OSCILLATOR | 1 |
| 6 | 2817A-200 | BUFFER | 1 |
| 7 | CMR03 | COMPARATOR | 2 |
| 8 | RNR55 | RESISTOR | 2 |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i = \text{ceil} ( u_i / upp_i )$$

**Explanation of symbols used:**

| | | |
|---|---|---|
| $n_i$ | = | Number of components (physical packages) required for i-th part |
| $u_i$ | = | Number of units (CAD symbols) required for i-th part |
| $upp_i$ | = | Number of units/package for i-th part (from Table 4.0) |
| ceil | = | Round up to nearest integer (e.g. 5.3 is rounded to 6) |

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: PENALTY EQUATIONS**

| | | |
|---|---|---|
| **AREA (sq.in.)** | = | Sum ( $n_i$ * $a_i$ ) + 15% for traces |
| **WEIGHT (gms)** | = | Sum ( $n_i$ * $w_i$ ) + 10% for solder |
| **POWER (mW)** | = | Sum ( $n_i$ * $p_i$ ) |
| **TEST TIME (sec)** | = | tMPBIT + tBILBO + tSCAN + tTERM |
| **DELAY (ns)** | = | 0.0 |

**Explanation of Symbols Used.**

| | | |
|---|---|---|
| Sum | = | Sum over parts in BIT Technique Parts Table (i=1 to 8) |
| $n_i$ | = | Number of component packages for i-th part in Parts Table<br>$n_i$ is calculated according to Component Determination Equations |
| $a_i$ | = | Area in sq.in. of i-th part in Parts Table |
| $p_i$ | = | Power dissipation in mW (typical) for i-th part in Parts Table |
| $w_i$ | = | Weight in grams of i-th part in Parts Table |
| tMPBIT | = | Microprocessor BIT execution time = 0.0347 sec |
| tBILBO | = | BILBO setup time = 0.000012 sec |
| tSCAN | = | SCAN setup time = 0.000012 sec |
| tTERM | = | BIT termination time = 0.0000005 sec |

**BIT TECHNIQUE: OBIVCB**

**CATEGORY: BIBLIOGRAPHY**

1. V.R. Subramanyam, L.R. Stine, (TRW), "Design for Testability for Future Digital Avionics Systems", IEEE, 1986. Describes a Module Maintenance Node which is the basis for the Test Processor Node in the OBIVCB.

2. R. Frohwerk, "Signature Analysis: A New Digital Field Service Method", Hewlett-Packard, 1977. Includes a good tutorial on Linear Feedback Shift Registers.

3. LeBlanc, "LOCST-LSSD On Chip Self Test", IEEE Design & Test of Computers, 1984.

4. D. Bacht, "Understanding Signature Analysis", Electronics Test, Nov. 1982, pg 28.

5. B. Konemann, Joachim Mucha and Gunther Zwiehoff, "Built-in Logic Block Observer", IEEE Test Conference, Cherry Hill, NJ, 1979.

# BUILT-IN LOGIC BLOCK OBSERVER (BILBO)

# BIT TECHNIQUE DATA PACKAGE

## CATEGORY: SHORT DESCRIPTION / DIAGRAM

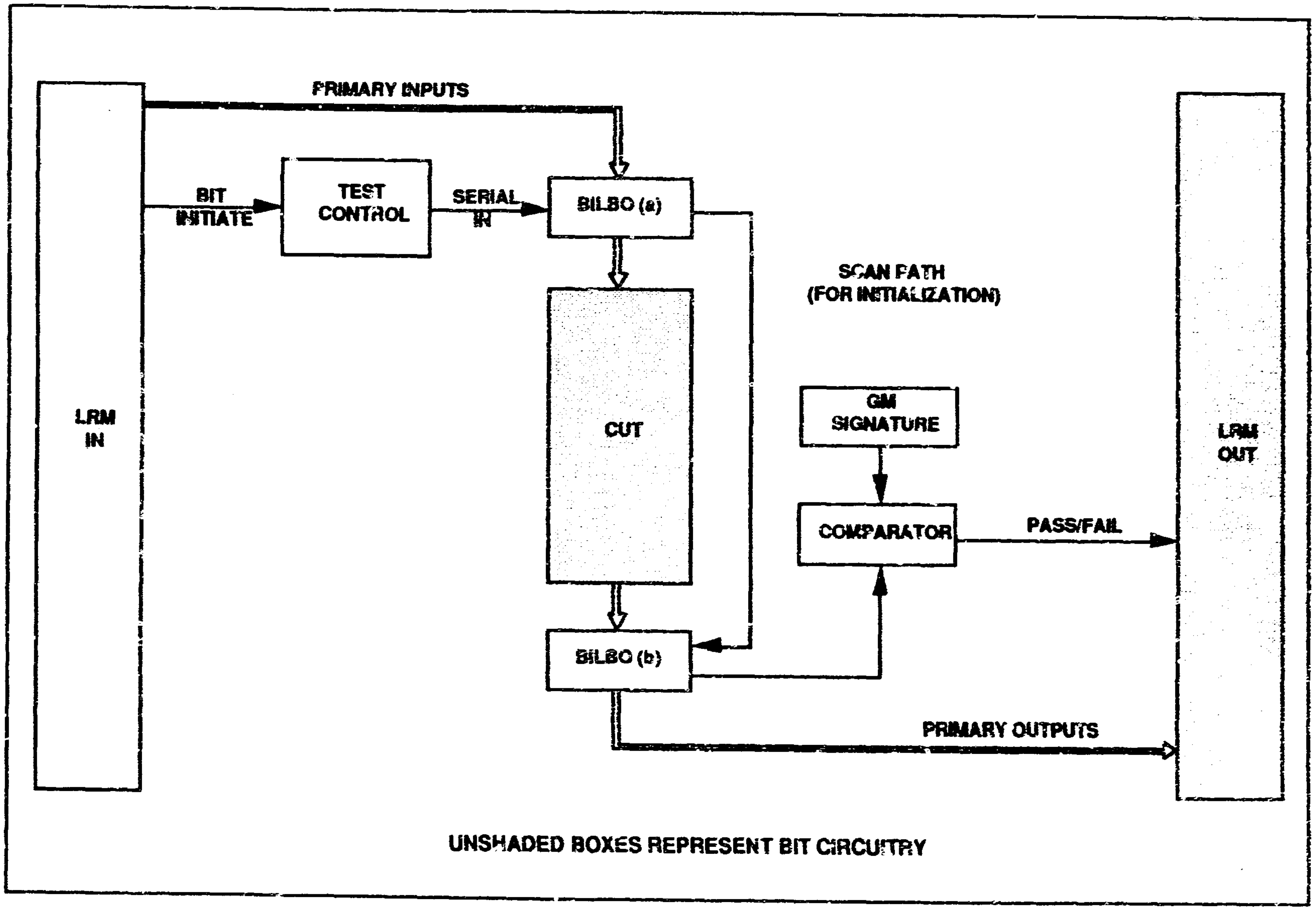
A Built-In Logic Block Observer (BILBO) is a multifunctional circuit which can be configured in any one of the following modes of operation:

1. Latch
2. Shift register
3. Multiple Input Signature Register (MISR) or Pseudorandcm Pattern Generator (PRPG)
4. Reset the BILBO

The configuration is selected by 2 mode control bits. BILBO circuitry can be used to perform signature analysis using a pseudorandom pattern generator (PRPG) and a multiple input signature register (MISR) which is an effective method of testing complex digital circuitry. The foundation of this Built-In-Test (BIT) technique is built on the fact that for a given sequence of stimuli, a fault-free Circuit Under Test (CUT) will output a particular digital stream. Using the data compression techniques of signature analysis, an output signature accumulated can be saved in a MISR. At the conclusion of the application of a given set of stimuli, the contents of the MISR are then compared against the known good signature. Initialization of the CUT and the BIT circuitry must be done prior to the execution of this technique. By utilizing the BILBO circuit, initialization can be easily achieved using scan-path techniques. The versatility of BILBO allows for the combination of two BIT techniques—SCAN and PRPG/MISR. This combination is especially useful when testing combinational logic stages that are separated by latches. Some of these latches are replaced with BILBO circuits. During normal operation, the BILBO functions as an ordinary latch.

During test initialization, each BILBO is configured as a serial shift register, and an initial seed value is shifted into the circuit. The BILBOs are then configured into a PRPG and MISR pairs. This allows for complete testing of the CUT.

Figures 1 and 2 provide Level I and II Block Diagrams for this BIT technique.



274_844_66

Figure 1 Level I Block Diagram For BILBO BIT Technique

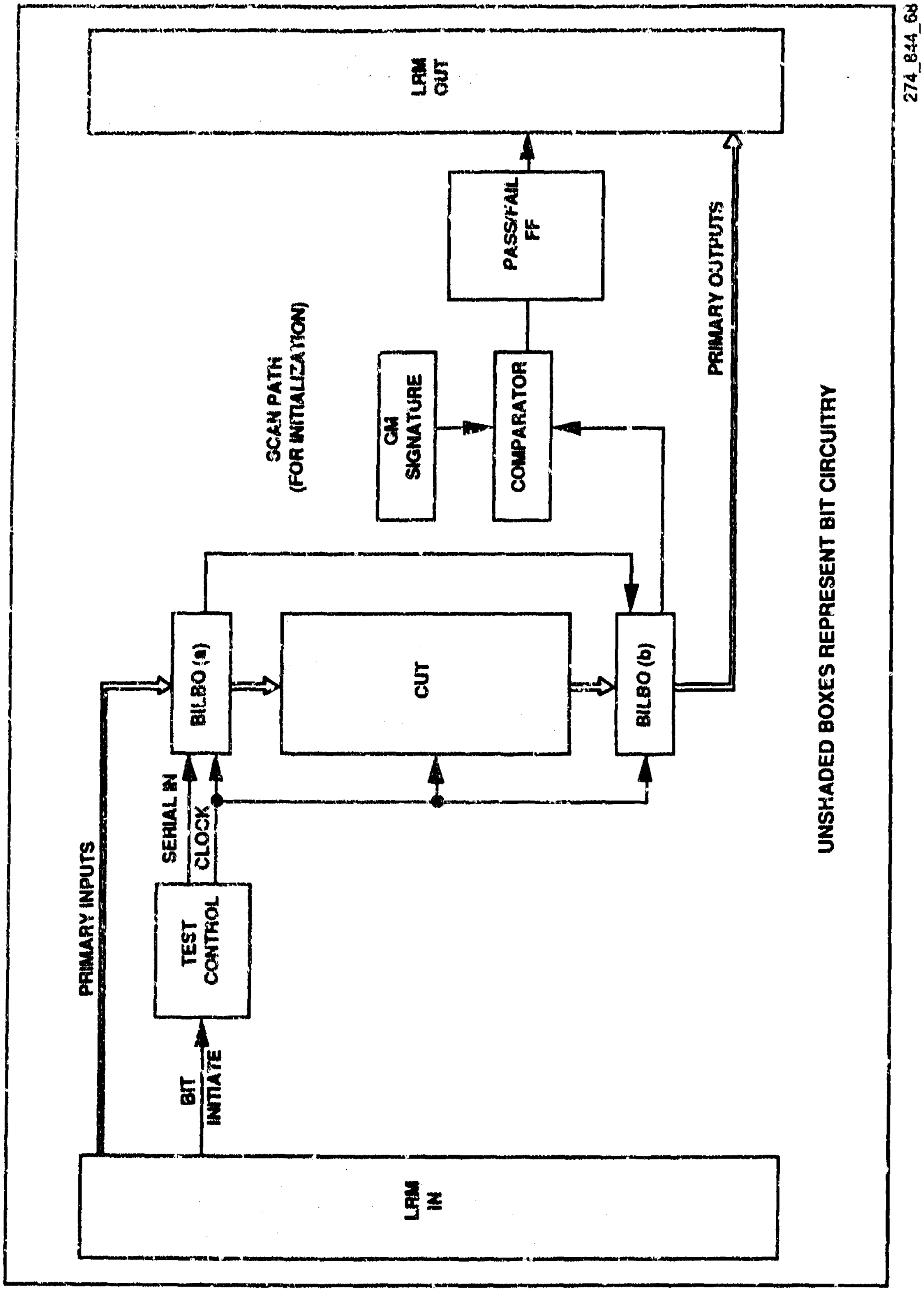


Figure 2 Level II Block Diagram For BILBO BIT Technique

**BIT TECHNIQUE: BUILT-IN LOGIC BLOCK OBSERVER (BILBO)**

**CATEGORY: FLOW CHART / DESCRIPTION**

Figure 3 shows the Flow Chart for this BIT technique.

1. Circuit Under Test (CUT) is powered up.

2. An initiate BIT signal is generated. Reset the PASS/FAIL Flip-Flop to PASS.

3. Using the BILBOs as shift registers, an initial seed pattern is scanned into the BILBO registers.

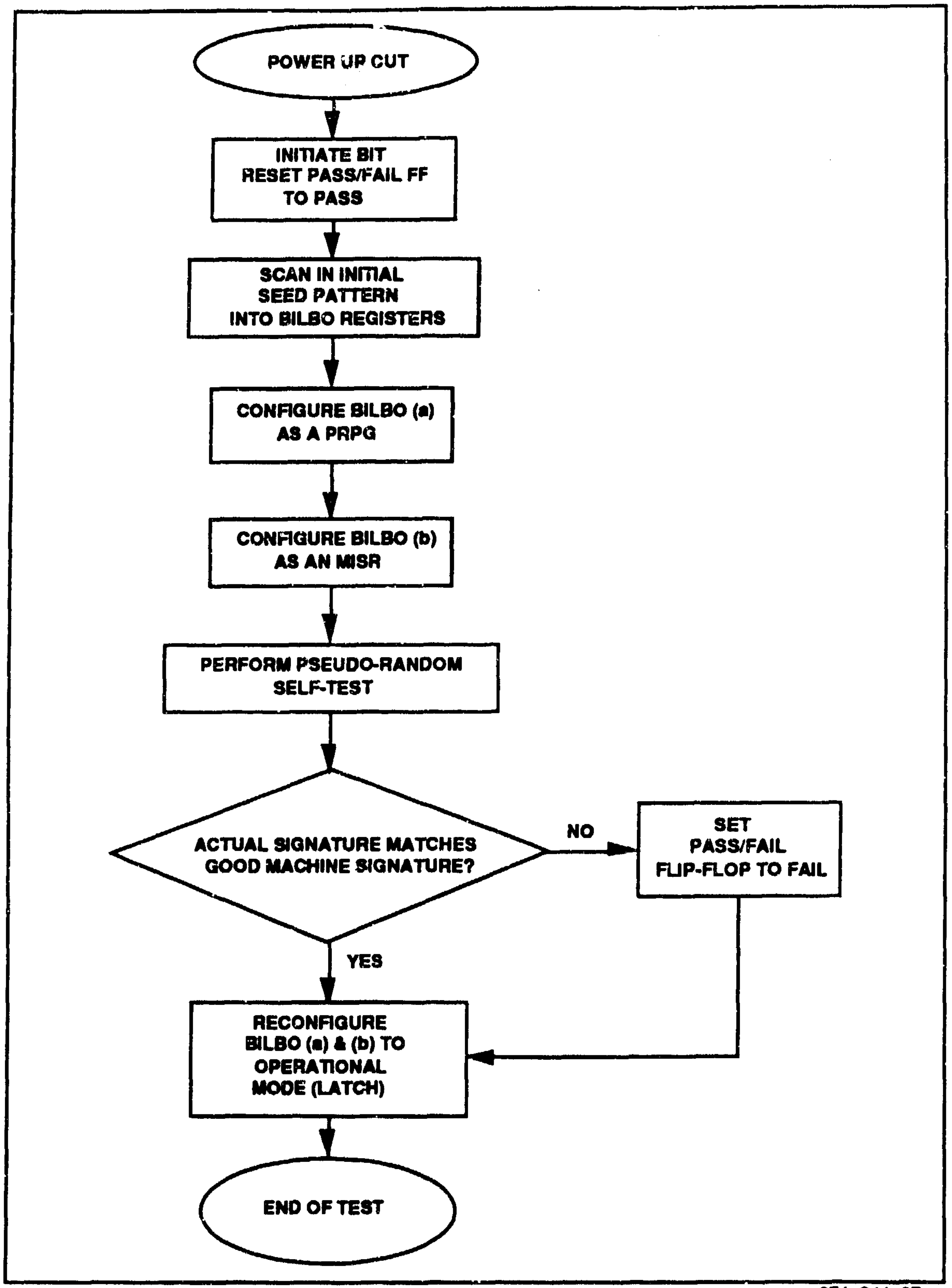
4. BILBO (a) is configured as a Pseudorandom Pattern Generator (PRPG) and BILBO (b) is configured as a Multiple Input Signature Register (MISR).

5. Random testing is performed.

6. BILBO (b) now contains the CUT signature which is fed into a comparator.

7. If the CUT signature does not match the Good Machine (GM) signature, set the PASS/FAIL Flip-Flop to FAIL. If the CUT signature matches the GM signature, the test passed.

8. BILBOs are reconfigured to function as latches.



274_844_67

Figure 3 BIT Sequence Flow Chart For Built-In Logic Block Observer (BILBO) BIT Technique

## BIT TECHNIQUE: BUILT-IN LOGIC BLOCK OBSERVER (BILBO)

## CATEGORY: ADVANTAGES

BILBO provides the following advantages to the circuit designer:

1. One circuit design can be used for multiple functions, giving the advantages of commonality (custom integrated circuit design can be made and used in many places).

2. The versatility of BILBO allows the designer to combine the advantages of SCAN techniques with PRPG/MISR techniques. There is a considerable reduction in test result storage achieved by signature analysis which provides data compaction algorithm.

3. The test data are gathered at the rated internal speed of the integrated circuits. No additional test clock is needed. The data sampling and compaction are performed completely inside the CUT.

4. A much higher failure detection rate is achieved when compared to other techniques such as transition counting. This is an efficient and compact tool for monitoring internal test points on complex digital I.C.s. Loss of fault coverage incurred using signature analysis is 5% for 8-15 bit MISRs, 1% for 16-23 bit MISR, and 0% for MISRs of 24 or more BITS (per MIL-STD-885 procedure 5012.1).

5. Minimal software support is required.

## CATEGORY: DISADVANTAGES

BILBO provides the following disadvantages to the circuit designer:

1. BILBO must be incorporated as part of the original design of the CUT

2. BILBO modules are more complex than the latches they replace. This results in additional circuitry.

3. Limited control of test vector sequencing which is more effective with high amount of combinational logic circuitry.

4. Circuit throughput delay will increase since BILBO is used as an input or output register.

## BIT TECHNIQUE: BUILT-IN LOGIC BLOCK OBSERVER (BILBO)

## CATEGORY: ATTRIBUTES

1. CONCURRENCY

   - Non-concurrent

2. TECHNOLOGY

   - Digital

3. CUT MICROPROCESSOR REQUIRED?

   - No

4. CUT INTERNAL DESIGN REQUIRED?

   - Yes

5. AREA PENALTY

   - BILBO latches are more complex and take up more area than the conventional latches they replace.
   - Some test control logic is also required.
   - The comparator which compares the good machine signature to the actual signature takes up area.

6. WEIGHT PENALTY

   - Proportional to the weight of the test control logic, the additional circuitry of the BILBO module, and the comparator.

7. POWER PENALTY

   - Proportional to the power dissipated by the test control logic, the additional circuitry of the BILBO module, and the comparator.

**BIT TECHNIQUE: BUILT-IN LOGIC BLOCK OBSERVER (BILBO)**

**CATEGORY: ATTRIBUTES, Contd**

8. TIMING PENALTIES

- Test time – equal to the number of test patterns divided by the application rate.
- Throughput delay – operational speed is slowed due to the propagation delay through the BILBO cells.

9. RELIABILITY IMPACT

- Failure rate increase due to Test Control Logic, the BILBO module, and the comparator.

10. CONCEPTUAL COMPLEXITY

- Circuit design is moderate in complexity.

11. HARDWARE/SOFTWARE/FIRMWARE

- Hardware

12. DESIGN COST

- All components used are readily available at low cost.
- Hardware design and debug is minimal.

13. MEMORY REQUIREMENTS

- Minimal, may be required to store test control routine and seeds.

14. IS BIT CIRCUITRY SELF TESTABLE?

- Yes

15. STAND-ALONE (SELF-CONTAINED) BIT?

- Yes

16. NOTES

- None

## BIT TECHNIQUE: BUILT-IN LOGIC BLOCK OBSERVER (BILBO)

## CATEGORY: DEFAULT DESIGN

This section describes the sequence of events for the BILBO BIT technique default design. Refer to Figures 4 and 5 for the Default Design schematic.

1. At the beginning of the test, test TEST_INITIATE signal goes low and triggers the following:

   a. Presets U1A test control flip-flop and enables TEST_CLOCK signal to activate counters and BILBOs.

   b. While the TEST_INITIATE signal holds low, the next rising edge of the TEST_CLOCK signal loads zeroes into the synchronous 4-bit binary counters U4, U5, and U6.

   c. The _Q of U1A test control flip-flop disables U10A, U10B, U10C, U10D, U11A, U11B, U11C, and U11D (and gates). This disables the normal inputs to BILBO #1 and maintains all zero inputs to BILBO #1 during testing.

   d. The _Q of U1A test control flip-flop enables U9 8-bit identity comparator. The Q of U1A test control flip-flop disables U12A and U12B octal buffers and places the outputs of CUT into high impedance

   e. Clears U2A and U2B flip-flops which sets B1 = '0' and B2 = '0'. This puts the BILBOs into Scan Mode.

   f. Clears the Pass/Fail flip-flop U1B to PASS.

2. After the TEST_INITIATE signal goes back to high, the TEST_CLOCK signal starts incrementing counters U4, U5, and U6. During the first sixteen clock cycles U5/QA outputs bit sequence 00000000 and 00000001. While in Scan Mode, BILBO #2 scans in 00000000 and BILBO #1 scans in 00000001 accordingly.

3. As soon as the initialization is completed, U2A flip-flop is set by QA of U5 counter to change B1 from '0' to '1' and stays latched to '1' during the testing. B1 = '1' and B2 = '0' puts the CUT into PRPG/MISR test mode.

**BIT TECHNIQUE: BUILT-IN LOGIC BLOCK OBSERVER (BILBO)**

**CATEGORY: DEFAULT DESIGN, Contd**

4. After 255 TEST_CLOCK cycles (2^8 - 1 = 255), U4/RCO and U6/QA are both high and indicates that the end of the PRPG/MISR test mode. These two signals are ended to trigger the following:

   a. Clears U1A test control flip-flop and disconnects TEST_CLOCK signal to counters and BILBOs.

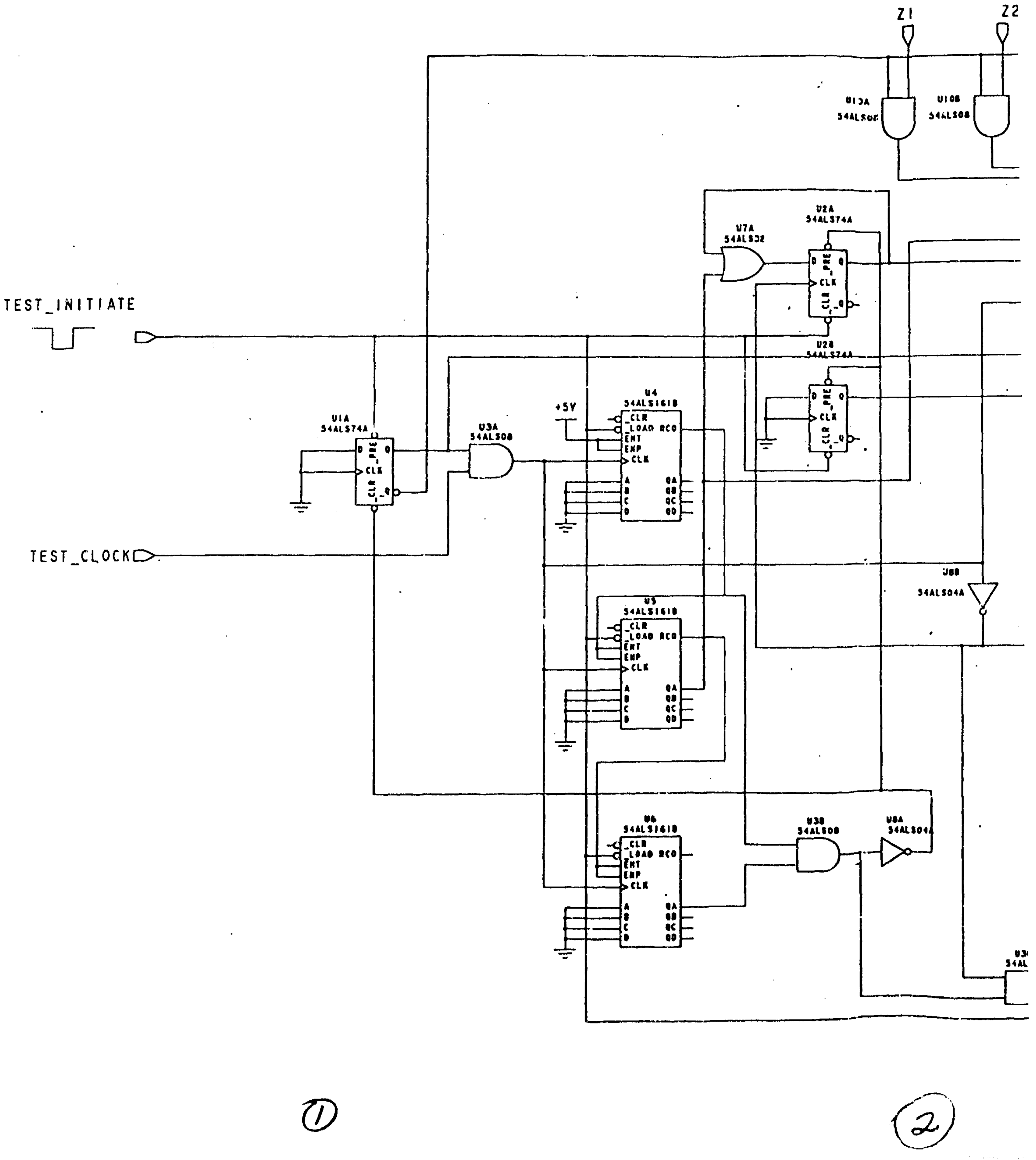
   b. Enables U10A< U10B, U10C, U10D, U11A, U11B, U11C, and U11D (and gates). This allows normal inputs flow to CUT.

   c. Disables U9 8-bit identity comparator and enables U12A and U12B octal buffers for the outputs of the CUT.

   d. Presets U2A and U2B flip-flops which sets B1 = '1' and B2 = '1'. This puts the BILBOs into Latch Mode.

   e. Clocks in the result of the comparison of Good Machine Response and test signature. This result is registered by the Pass/Fail flip-flop U1B.

5. The circuit resumes normal operation and it will go into test mode when the TEST_INITIATE signal goes low again.


Z1
Z2
U10A
54ALS08
U10B
54ALS08
U2A
54ALS74A
U7A
54ALS32
D
_PRE
Q
CLK
_CLR
_Q
TEST_INITIATE
U2B
54ALS74A
U4
54ALS161B
+5V
U1A
54ALS74A
U3A
54ALS08
_CLR
_LOAD
RCO
ENT
ENP
CLK
A
B
C
D
QA
QB
QC
QD
TEST_CLOCK
U8B
54ALS04A
U5
54ALS161B
U6
54ALS161B
U3B
54ALS08
U8A
54ALS04A
U3
54AL
1
2

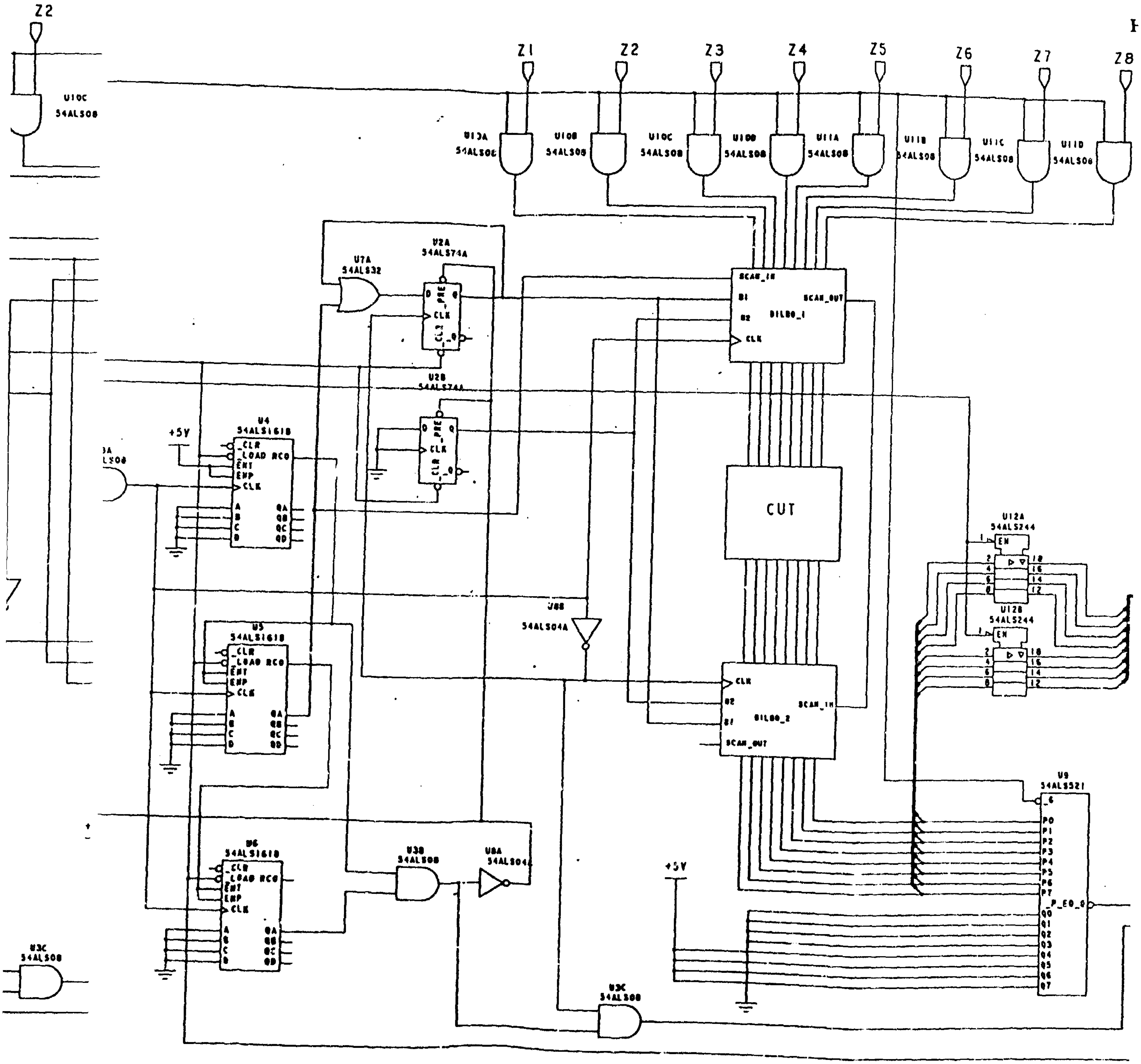


Figure 4 BILBO – Default Design

2

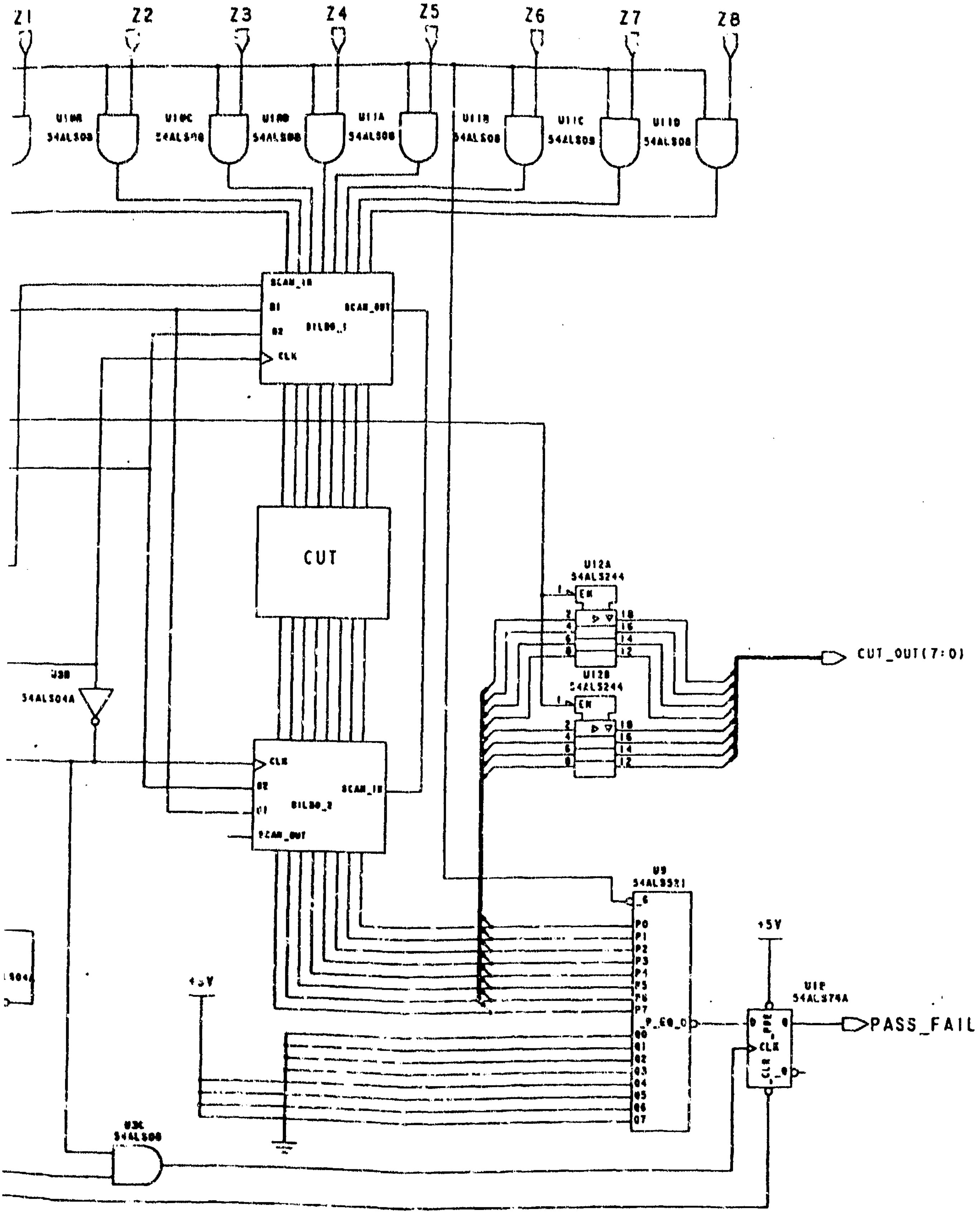
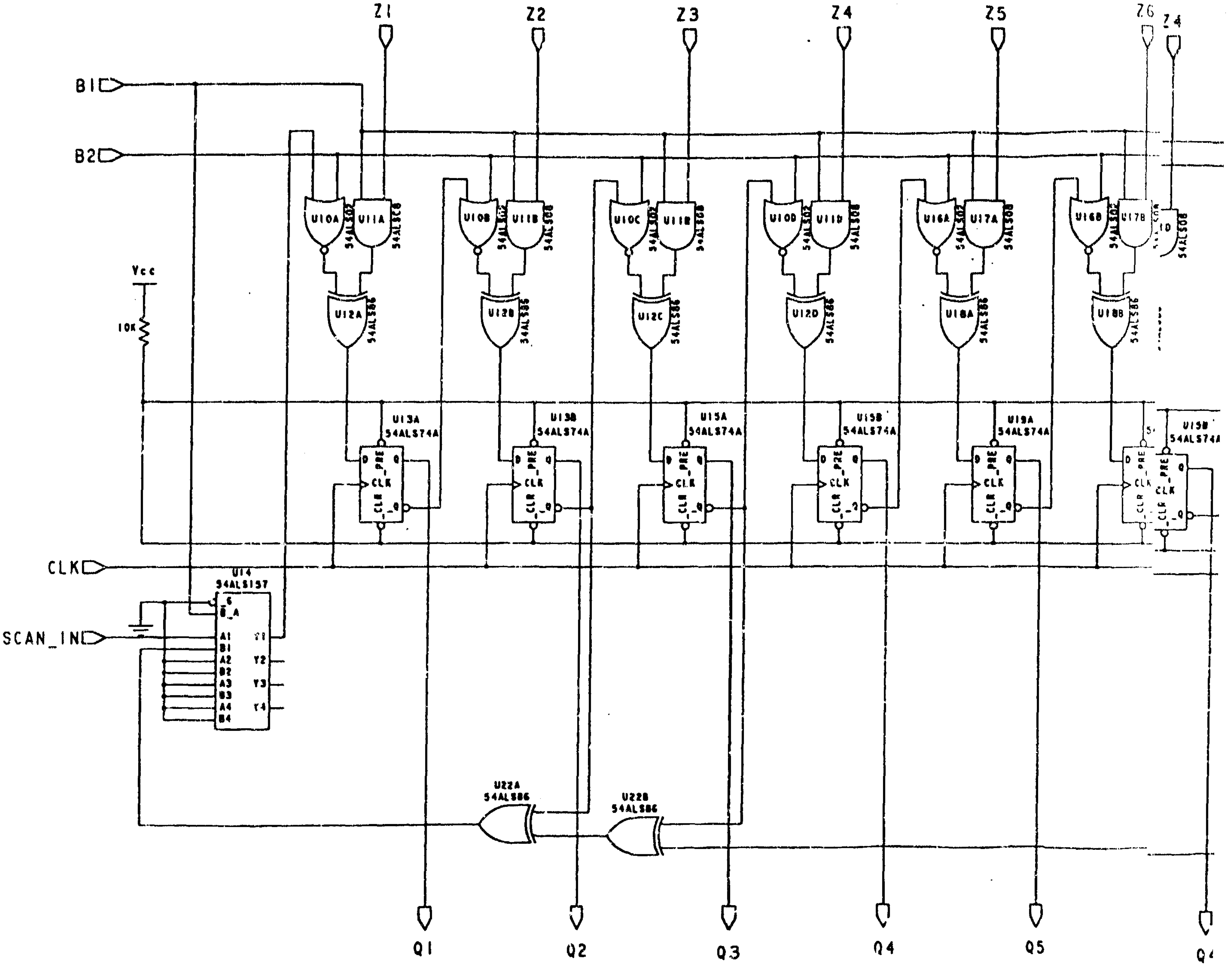
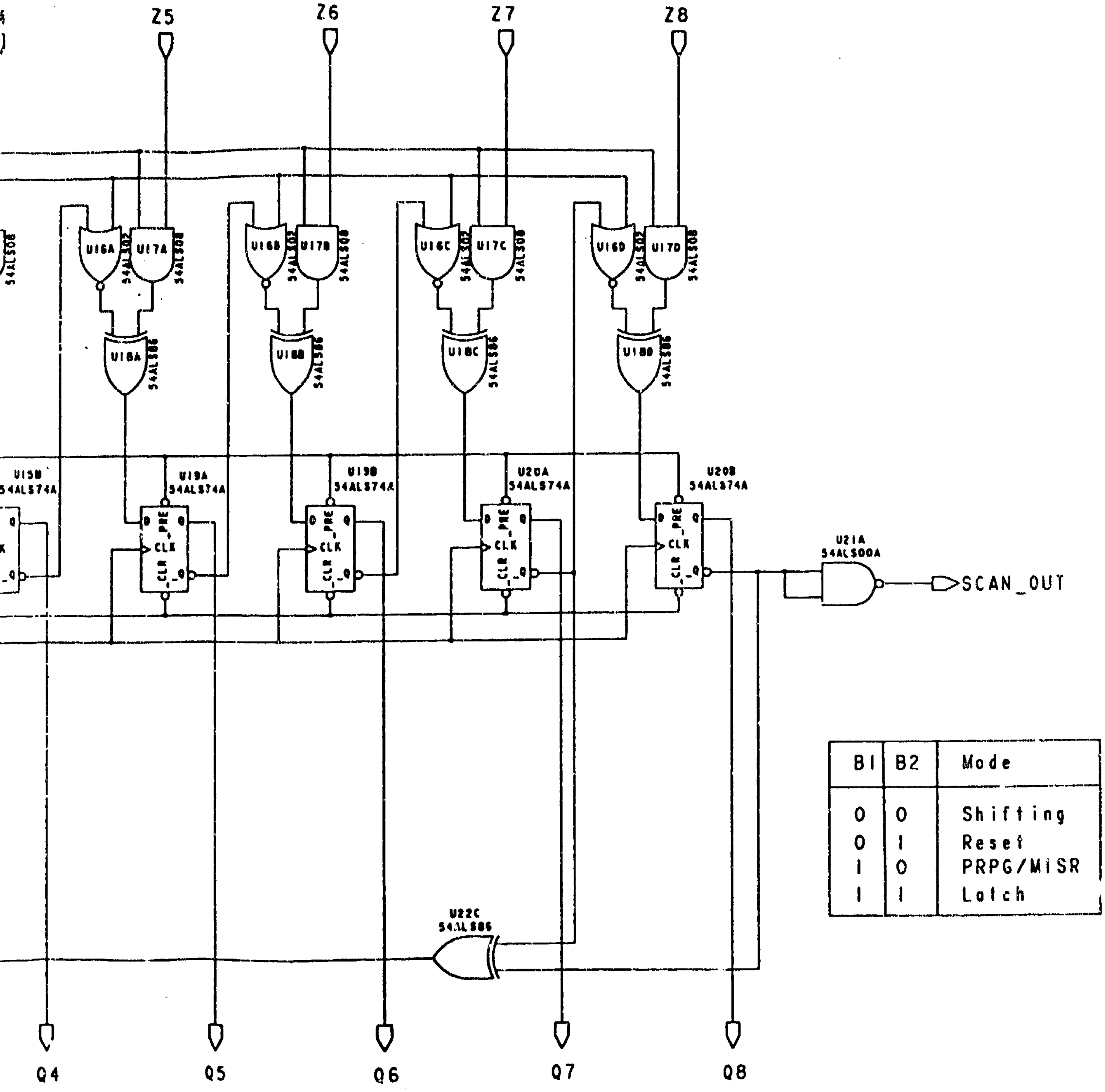


Figure 4 BILBO – Default Design


Z1
Z2
Z3
Z4
Z5
Z6
Z4
B1
B2
U10A
U11A
54ALS02
54ALS08
U10B
U11B
U10C
U11C
U10D
U11D
U16A
U17A
U16B
U17B
U11D
U12A
U12B
U12C
U12D
U18A
U18B
54ALS86
Vcc
10K
U13A
54ALS74A
U13B
U15A
U15B
U19A
D
PRE
Q
CLK
CLR
CLK
U14
54ALS157
G
B_A
SCAN_IN
A1
B1
A2
B2
A3
B3
A4
B4
Y1
Y2
Y3
Y4
U22A
54ALS86
U22B
54ALS86
Q1
Q2
Q3
Q4
Q5

1



| B1 | B2 | Mode |
|---|---|---|
| 0 | 0 | Shifting |
| 0 | 1 | Reset |
| 1 | 0 | PRPG/MISR |
| 1 | 1 | Latch |

Figure 5 BILBO – Default Design

## BIT TECHNIQUE: BUILT-IN LOGIC BLOCK OBSERVER (BILBO)

## CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM

Figure 6 shows the BIT Technique Insertion Diagram for this BIT technique.

BIT ELEMENT 1:

1. Connect the control gates and Flip-Flop to the control counters.

BIT ELEMENT 2:

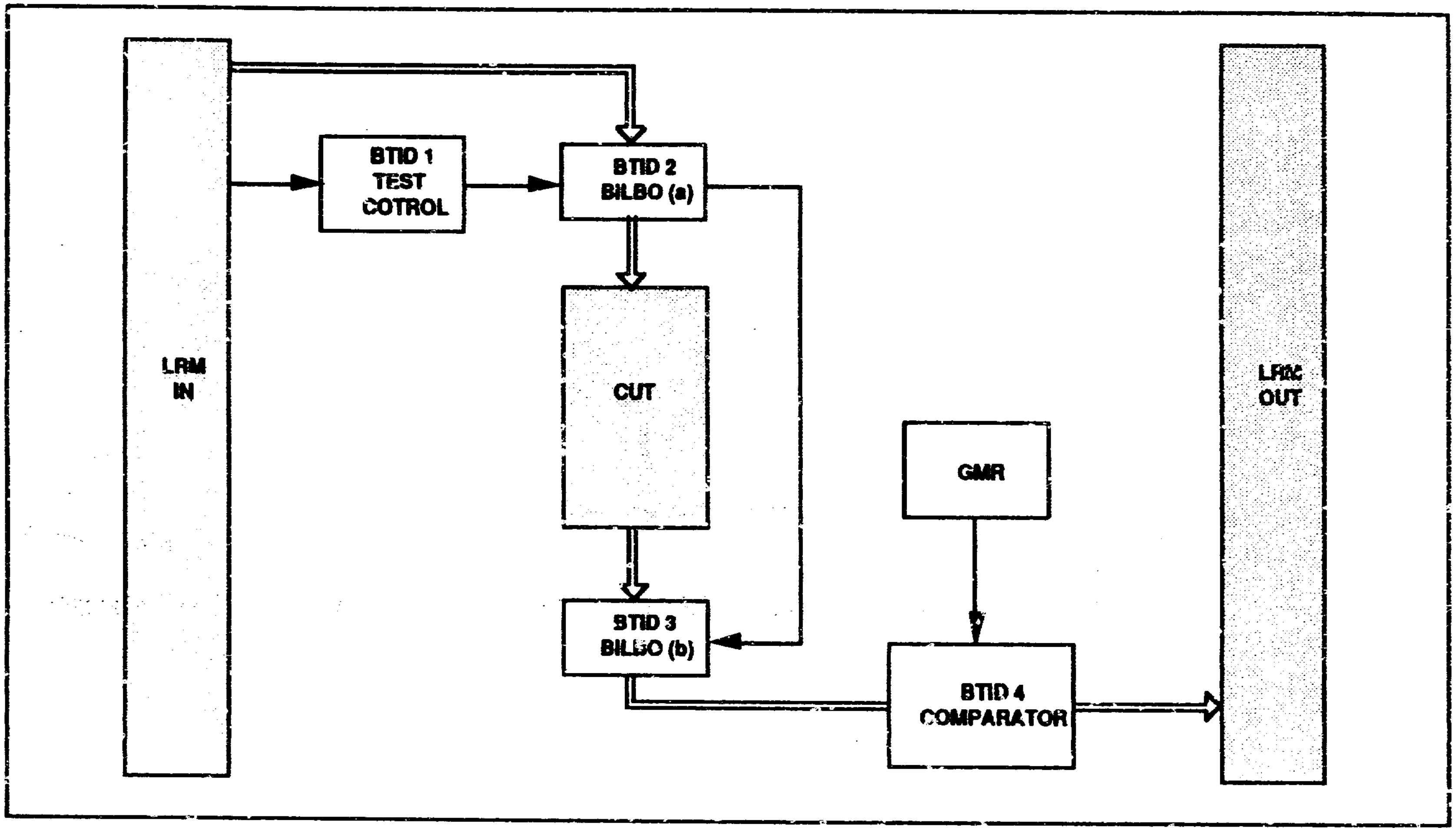
1. Connect the seed counter output to the inputs of BILBO (a).
2. Connect the output of BILBO (a) to the CUT.

BIT ELEMENT 3:

1. Connect the CUT outputs to the input of BILBO (b).

BIT ELEMENT 4:

1. Connect the output of BILBO (b) to the comparator and the MUX inputs.
2. Connect the output of the comparator to the input of the D input of the PASS/FAIL Flip¡Flop.
3. Connect the reset signal to the PASS/FAIL Flip-Flop CLR signal line.
4. Connect the MUX output to the LRM output.



274_844_69

Figure 6 BIT Technique Insertion Diagram-Top Level

## BIT TECHNIQUE: BUILT-IN LOGIC BLOCK OBSERVER

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | Number of CUT inputs | none |
| $v_2$. | Number of CUT outputs to test | none |
| $v_3$. | Maximum propagation delay through CUT | ns |
| $v_4$. | CUT initialization time | sec |
| $v_5$. | Internal Design OK? (Suitability Attribute) | 0 if YES; 1 if NO |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 54ALS00A | NAND GATES | 2 |
| 2 | 54ALS02 | NOR GATES | $v_1 + v_2$ |
| 3 | 54ALS04B | INVERTER | 2 |
| 4 | 54ALS08 | AND GATES | $(v_1+3) + v_1 + v_2$ |
| 5 | 54ALS32 | OR GATES | 1 |
| 6 | 54ALS74A | FLIP-FLOP | $3 + v_5*(v_1 + v_2) + 1$ |
| 7 | 54ALS86 | EXCLUSIVE-OR | $(v_1+3) + (v_2+3)$ |
| 8 | 54ALS157 | SELECTOR/MUX | 2 |
| 9 | 54ALS161B | COUNTER | ceil ( $\log_2(p) / b$ ) |
| 10 | 54ALS244A | BUFFER | ceil ( $v_2 / b$ ) |
| 11 | 54ALS521 | COMPARATOR | ceil ( $v_2 / b$ ) |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i = \text{ceil} ( u_i / upp_i )$$

**Explanation of symbols used:**

| | | |
|---|---|---|
| $n_i$ | = | Number of components (physical packages) required for i-th part |
| $u_i$ | = | Number of units (CAD symbols) required for i-th part |
| $upp_i$ | = | Number of units/package for i-th part (from Table 4.0) |
| ceil | = | Round up to nearest integer (e.g. 5.3 is rounded to 6) |
| b | = | Number of data bits for part (from Table 4.0) |
| p | = | Number of test patterns = $2^{v1} - 1 + v_1 + v_2$ |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |

**BIT TECHNIQUE: BUILT-IN LOGIC BLOCK OBSERVER (BILBO)**

**CATEGORY: PENALTY EQUATIONS**

|  |  |  |
|---|---|---|
| **AREA (sq.in.)** | = | Sum ( $n_i * a_i$ ) + 15% for traces |
| **WEIGHT (gms)** | = | Sum ( $n_i * w_i$ ) + 10% for solder |
| **POWER (mW)** | = | Sum ( $n_i * p_i$ ) |
| **TEST TIME (ns)** | = | 0 (BILBO is a |
| **DELAY (ns)** | = | tAND + 2*( tAND + tOR + $v_5$ * tFF) + tBUFFER |

**Explanation of Symbols Used.**

|  |  |  |
|---|---|---|
| Sum | = | Sum over parts in BIT Technique Parts Table (i=1 to 11) |
| $n_i$ | = | Number of component packages for i-th part in Parts Table<br>$n_i$ is calculated according to Component Determination Equations |
| $a_i$ | = | Area in sq.in. of i-th part in Parts Table |
| $p_i$ | = | Power dissipation in mW (typical) for i-th part in Parts Table |
| $w_i$ | = | Weight in grams of i-th part in Parts Table |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |
| M | = | Design margin = 1.2 |
| p | = | Number of test patterns = $2^{v1} - 1 + v_1 + v_2$ |
| tPRPG | = | PRPG time = tFF + 4tXOR + tMUX + tNOR |
| tAND | = | Max delay for AND GATES from Table 4.0 |
| tOR | = | Max delay for OR GATES from Table 4.0 |
| tNOR | = | Max delay for NOR GATES from Table 4.0 |
| tXOR | = | Max delay for EXCLUSIVE-OR from Table 4.0 |
| tMUX | = | Max delay for SELECTOR/MUX from Table 4.0 |
| tCOMP | = | Max delay for COMPARATOR from Table 4.0 |
| tFF | = | Max delay for FLIP-FLOP from Table 4.0 |
| tBUFFER | = | Max delay for BUFFER from Table 4.0 |

**BIT TECHNIQUE: BUILT-IN LOGIC BLOCK OBSERVER (BILBO)**

**CATEGORY: BIBLIOGRAPHY**

BILBO - Built-In Logic Block Observation Techniques
79 - Koenemann, Mucha, Zwieoff - 1979 IEEE Test Conference

81 - Segers - 1981 IEEE Test Conference
A Self-Test Method for Digital Circuits

STUMPS - Self Testing of Multi Chip Logic Modules
82 - Bardell, McAnney - 1982 IEEE Test Conference

83 - Komonysky - Electronics 1983 -
Synthesis of Techniques Creates Complete System Self-Test

84 - Butt, El-ziq - 1984 International Test Conference -
Impact of Mixed-Mode Self-Test on Life Cycle Cost of VLSI Based Designs

85 - Bhavsar - 1985 International Test Conference
"Concatenabie Polydividers": Bit-Sliced LFSR Chips for Board Self-test

85 - Kraniewski, Albicki - "Self-testing Pipelines"

THIS PAGE INTENTIONALLY LEFT BLANK

# ERROR DETECTION AND CORRECTION CODES (EDCC)

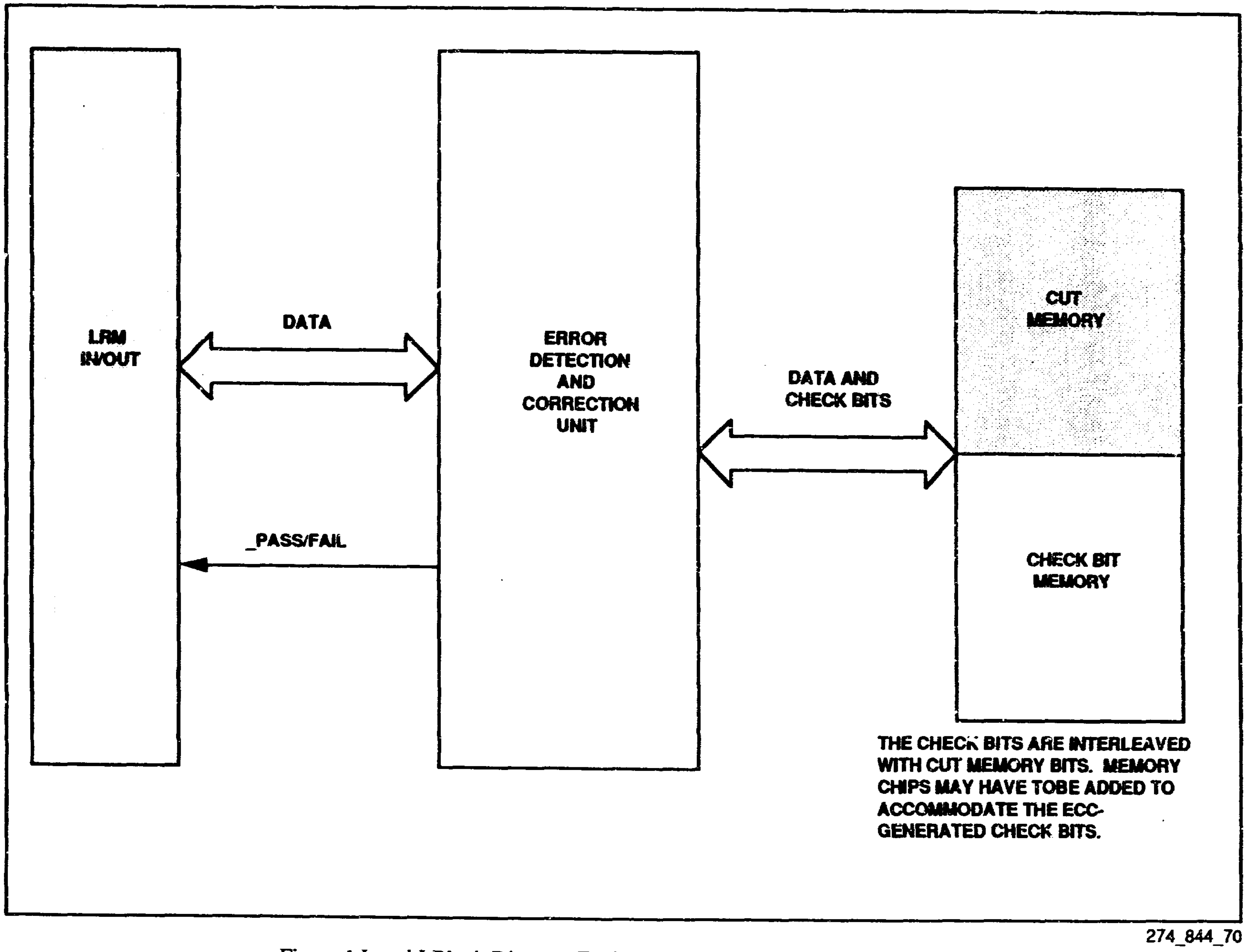
## BIT TECHNIQUE DATA PACKAGE

**CATEGORY: SHORT DESCRIPTION / DIAGRAM**

Used as a concurrent Built-In-Test (BIT) Technique, Error Detection and Correction units provide greater memory system reliability through their ability to detect and correct memory errors. Using similar techniques to parity, Hamming codes generate extra encoding bits which are appended to the data word which is to be transmitted or stored in memory. When the data and extra encoding bits are read from memory, a new set of code bits, read check bits, are generated. Comparison is made between various bit combinations of the data read from memory by the use of exclusive-or gates. The result of the comparison, called the syndrome word, contains information to determine if an error has occurred. Unlike parity, the syndrome word also contains information to indicate which bit is in error. After decoding this information, a flag can be set to indicate if an error occurred. Error correction with single bit errors is accomplished by inverting the bit in error. Identification of the bit in error is decoded from the syndrome word to provide the binary value of the bit position.

During memory write operations, write check bits are generated and stored in memory appended to data memory. This data is used on subsequent memory read operations to detect and correct errors.

Figures 1 and 2 show the Level I and II Block Diagrams for this BIT technique.



Figrue 1 Level I Block Diagram For Error Detection And Correction Codes

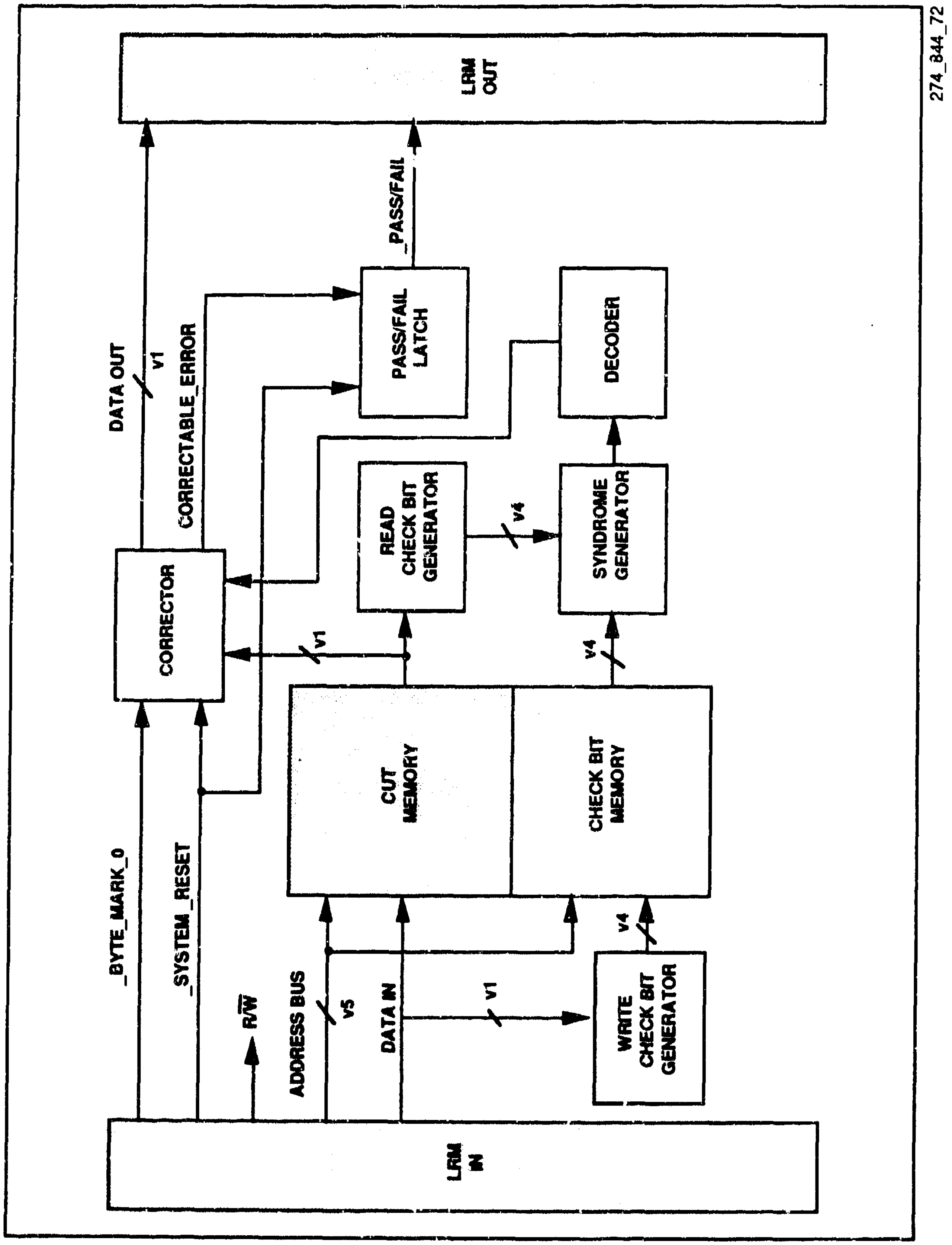


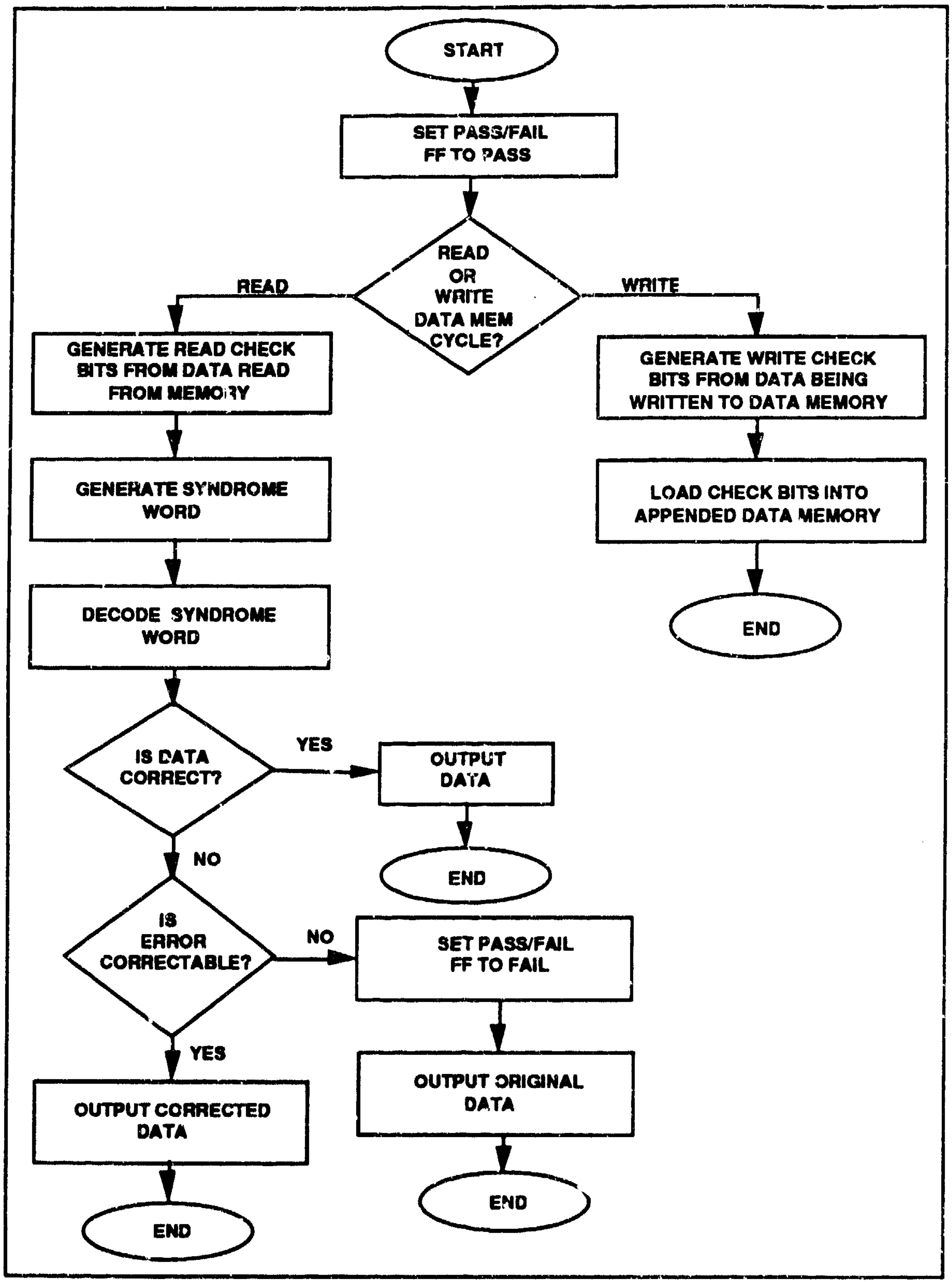
Figure 2 Level II Block Diagram For Error Detection And Correction Codes

## BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES

## CATEGORY: FLOW CHART / DESCRIPTION

Figure 3 shows the Flow Chart for this BIT technique.

1. Set the PASS/FAIL Flip Flop to Pass.
2. Determine if memory cycle is a read or write.
3. If memory cycle is a read, generate read check bits from data out of memory. If memory cycle is a write, jump to Step 9.
4. Generate syndrome word by exclusive or operation of the fetched check bits and the regenerated check bits.
5. Decode the syndrome word to determine which bits are in error.
6. If no error is detected, send the data to the output.
7. If an error is detected and the "Correctable Error" flag is set, correct the detected error and send the corrected data word to the output. Skip Steps 8 through 10.
8. If the error is not correctable, set the PASS/FAIL Flip Flop to FAIL and output the original data.
9. Generate write check bits from data into memory of Circuit Under Test (CUT).
10. Load check bits into appended data memory used for storing the check bits.



274_844_71

Figure 3 BIT Sequence Flow Chart

**BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES**

**CATEGORY: ADVANTAGES**

1. Error detection and correction code (EDCC) can improve memory system reliability. Read or write errors produced in memory can be corrected using this technique resulting in better system reliability.

2. Relatively small amount of hardware required to use EDCC

3. All in one chips are available to accomplish error detection and correction.

4. Error Correction Code (ECC) chips can be cascaded for expanded word length.

5. If 1-bit wide memory chips are used, a whole chip failure can be tolerated.

**BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES**

**CATEGORY: DISADVANTAGES**

1. Efficiencies of single detect and single detect/single correct codes decreases as the number of data bits decreases. For example, an 80 bit data word requires only 8 check bits whereas an 8 bit system requires 5 check bits.

2. Possible large hardware requirement for interfacing with large memory circuits. Each ECC chip can handle 16 bits of the data word. For 80 bit data words, 5 ECC chips would have to be cascaded.

3. Some decrease in throughput due to error correction element delay. For high speed systems this delay (typically 67 nsec) could significantly slow down the system.

4. Requires addition of Random Access Memory (RAM) for check bits. Some memory configurations could double the number of RAM chips required.

5. Additional traces required may increase the complexity of the layout and board design.

6. If additional memory chips are needed, the amount of board space required increases.

## BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES

## CATEGORY: ATTRIBUTES

1. CONCURRENCY

- BIT circuit runs concurrently with the CUT.

2. TECHNOLOGY

- This BIT technique would typically be used for CUTs using DIGITAL hardware.
- Many highly integrated single chip ECCs are available off-the-shelf.
- If ECC integrated circuits are not available in a particular technology, ECC can be implemented using SSI/MSI or ASIC technology with an increased penalty in real estate and power.

3. CUT MICROPROCESSOR REQUIRED?

- NO. This BIT technique is used to detect and correct errors in read/write memories.

4. CUT INTERNAL DESIGN REQUIRED?

- YES. CUT needs to be modified to add memory for check bits.

5. AREA PENALTY

- Dependent on number of ECC chips needed. Each ECC chip can handle up to 16 data bits and 8 check bits. Therefore, minimal impact if word length is less than 16 bits.
- Dependent on number of added memory chips for check bits. This is a function of both width and depth of the memory. Could double the required area for memory chips.

6. WEIGHT PENALTY

- Weight penalty is dependent on the number of ECC and memory chips used which is dependent on the data word width and depth.

**BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES**

**CATEGORY: ATTRIBUTES, Contd**

7. POWER PENALTY

- Power dissipation is dependent on the number of ECC and memory chips used which is dependent on the data word width and depth.
- Can be reduced if low standby power memory chips are used.

8. TIMING PENALTY

- Increases machine cycle time by 67 nsec typically (for the I8206) per read cycle. In high speed systems the increased machine cycle time could slow down the system significantly.

9. RELIABILITY PENALTY

- Slight decrease of hardware reliability due to addition of a small number of memory and ECC chips. For large memories. system reliability is increased because the probability of a one bit error is greater than the probability of BIT circuit failure.

10. CONCEPTUAL COMPLEXITY

- Straightforward.

11. HARDWARE/SOFTWARE/FIRMWARE

- No software required at the board level.
- Hardware includes ECC, memory, and a few ALS logic chips.

12. DESIGN COST

- Minimal if ECC integrated circuits are used.

13. MEMORY REQUIREMENTS

- Isolated input and output port RAM is needed to store the check bits if the CUT memory word depth is not sufficient.

**BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES**

**CATEGORY: ATTRIBUTES, Contd**

14. BIT CIRCUITRY SELF-TESTABLE?

- Check bits are tested with data bits, but ECC chips themselves are not self-testable.

15. STAND-ALONE (SELF-CONTAINED) BIT?

- YES, if fault is in ECC integrated circuit, a correctable error flag will be detected unless fault is in error flag output.

16. NOTES

- None required.

## BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES

## CATEGORY: DEFAULT DESIGN

This section describes the Default Design for the Error Detection and Correction Codes (EDCC) BIT technique. Refer to Figure 4 for the Default Design schematic.

Functionally, the design uses an Error Detection and Correction Unit (EDCU) (U1) for detecting and correcting errors during memory operations, isolated port SRAM (U2 and U3) for storing check bit data, and a Flip Flop (FF) with NAND gate control (U4 and U5) for the BIT PASS/FAIL signal.

The Default Design shows the simple case of an 8 bit data bus system where data read from memory can be checked and corrected for single bit errors. For data written to memory, the Default Design generates the check bits and stores them in the check bit memory.

The Intel I8206 was chosen as the EDCU. It uses a modified Hamming code to detect and correct single bit errors during memory read cycles. Each chip can handle 8 or 16 bits of data and up to 8 check bits. Up to 5 chips can be cascaded to handle data paths of up to 80 bits.

A byte of data in memory that is known to be in error can be corrected and stored back into the same memory location by using a Read-Modify-Write cycle. The ERROR and CE (Correctable Error) outputs of the I8206 are used to determine if the present byte being read from memory is in error, and, if so, if it is correctable. If a correctable error is detected, a Read-Modify-Write cycle can be run to correct the error permanently. The Default Design does not implement this feature.

One of the deficiencies of the I8206 is that it uses two separate 16-pin busses, one to accept data from RAM (DI) and the other to deliver corrected data to the system bus (DO/WDI). This necessitates the use of isolated port RAM for data memory which is not as readily available as single port RAM. Also, the I8206 adds 57 nsec typically to the processor machine cycle time to generate the corrected data. This could seriously impact the speed of high speed systems. Other manufacturers make EDCU chips which get around these problems. Also, error detection can take place in parallel with the normal read cycle. The system would be slowed down only when a detected error is corrected.

## BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES

## CATEGORY: DEFAULT DESIGN, Contd

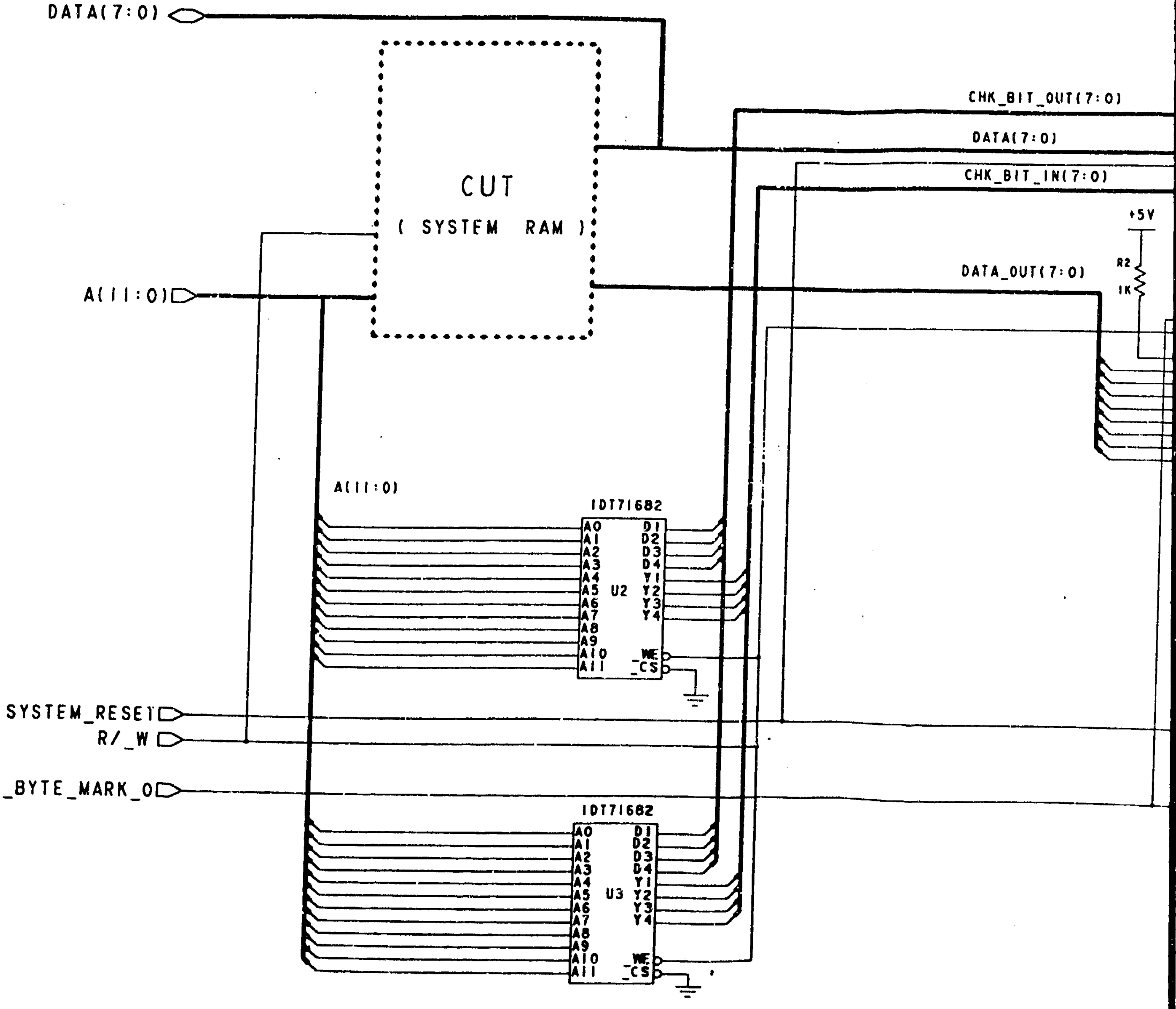
Because the I8206 EDCU has separate check bit in and out ports, isolated port RAM is also needed for storing the check bits. The SRAM chosen for this design was the Integrated Device Technologies IDT71682. It is arranged as a 4K x 4 bit RAM and two are required to store the 5 to 8 check bits necessary for 8 to 80 bit data paths.

A 54ALS74A D Flip Flop (U4) and a 54ALS00A NAND gate (U5) perform the PASS/FAIL latch function. The CE output of the EDCU is used to alert the system if an uncorrectable error has occurred.

System Interconnect: Three signals in addition to the Address Bus and the Data Bus are used to interface the BIT circuit to the system processor. The _BYTE_MARK_0 signal is normally low and goes high at the rising edge of R/_W. It is held high for the sum of the CUT data memory access time, the data in to CE valid time, and a safety factor. It is used to control the EDCU and to clock the CE signal in the PASS/FAIL FF. The _SYSTEM_RESET is used to set the PASS/FAIL FF to PASS and to write all zeros from the EDCU to the data bus during power initialization. R/_W is the standard read/write signal generated by the system processor to control memory operations.

Data Flow: During a read operation, data flows from the output port of the CUT data memory into the Data In (DI) port of the EDCU. Also, check bit data is read from U2 and U3 into the Check Bit In (CBI) port of the EDCU. Corrected data is placed on the system data bus through the EDCU Data Out/Write Data In (DO/WDI) port. If an uncorrectable error occurs, the CE output goes low and the PASS/FAIL FF is latched low at the falling edge of _BYTE_MARK_0.

During a write operation, data flows into both the CUT data memory input bus and the EDCU DO/WDI bus from the system data bus. Check bits are generated at the EDCU Check Bit Out (CBO) bus and loaded into the check bit memory (U2 and U3).


DATA(7:0)
CUT
( SYSTEM RAM )
CHK_BIT_OUT(7:0)
DATA(7:0)
CHK_BIT_IN(7:0)
+5V
R2
1K
DATA_OUT(7:0)
A(11:0)
A(11:0)
IDT71682
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
U2
D1
D2
D3
D4
Y1
Y2
Y3
Y4
_WE
_CS
SYSTEM_RESET
R/_W
_BYTE_MARK_0
IDT71682
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
U3
D1
D2
D3
D4
Y1
Y2
Y3
Y4
_WE
_CS

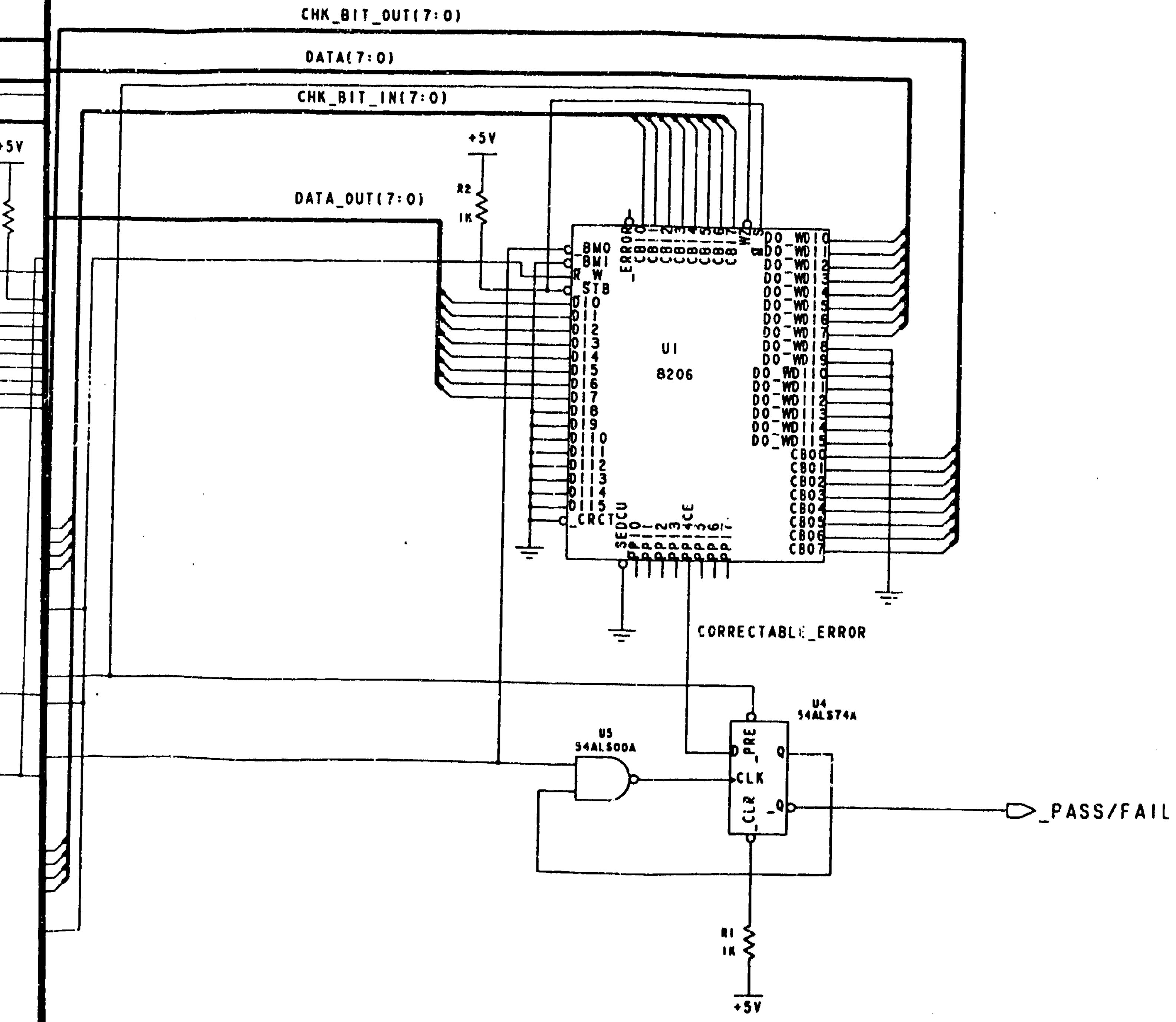


Figure 4 EDCC – Default Design

2

## BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES

## CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM

Figure 5 shows the BIT Technique Insertion Diagram for this BIT technique.

BIT ELEMENT 1, ERROR DETECTION AND CORRECTION UNIT (EDCU)

1. Connect _BYTE_MARK_0 and _SYSTEM_RESET from the LRM to the EDCU _BM0 and _WZ lines, respectively. The _BYTE_MARK_0 and _SYSTEM_RESET are signals generated by the system _BYTE_MARK_0 is a normally low signal which goes high at the rising edge of the system R/_W signal. It must be held high for a minimum of the sum of the check bit memory access time (20 to 70 nsec depending on the device speed option selected) plus the time for the I8206 Correctable Error (CE) output to become valid from the data/check bit valid time plus a safety factor. _SYSTEM_RESET is the same reset line that is used to initialize the rest of the system.

2. Connect the DATA OUT bus of the CUT memory to the DATA IN (DI) bus of the EDCU.

3. Connect the CHECK BIT IN and CHECK BIT OUT busses of BIT Element 2, Check BIT Memory, to the corresponding EDCU busses, CBI and CBO.

4. Connect the LRM DATA BUS to the CUT DATA IN bus and to the EDCU DATA OUT/WRITE DATA IN bus.

5. Connect the LRM R/_W line to the EDCU R/_W line.

6. Connect the PPI4CE, CORRECTABLE_ERROR, line from the EDCU to BIT Element 3, PASS/FAIL LATCH.

BIT ELEMENT 2, CHECK BIT MEMORY

1. Connect the LRM ADDRESS BUS to the Check BIT Memory ADDRESS BUS.

2. Connect the LRM R/_W line to the _WE line of the Check BIT Memory.

BIT ELEMENT 3, PASS/FAIL LATCH

1. Connect the _PASS/FAIL output of the PASS/FAIL LATCH to the LRM _PASS/FAIL input.

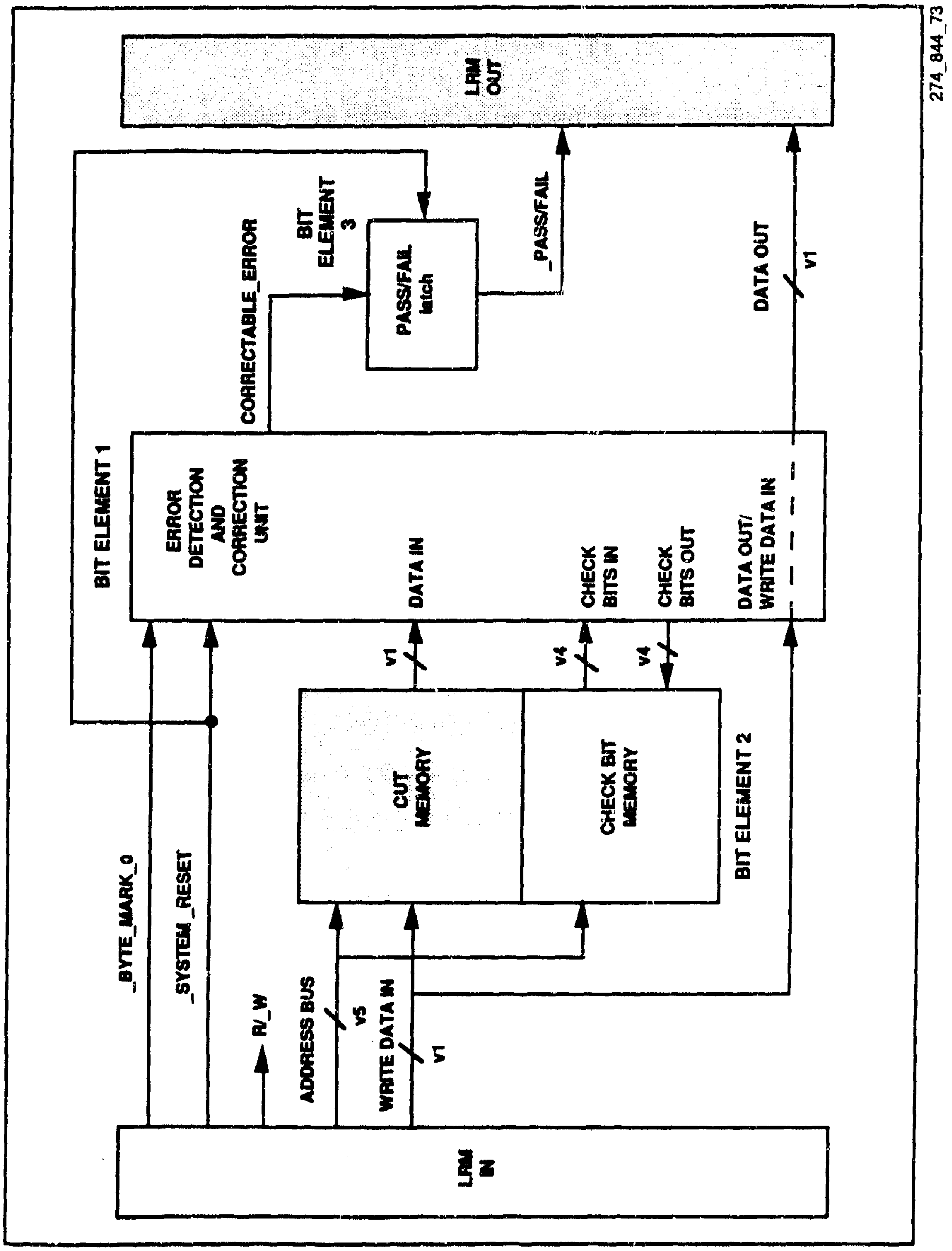


Figure 5 BTID Block Diagram Utilizing Error Detection And Correction Codes

## BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | Number of CUT inputs | bits |
| $v_2$. | #CUT memory locations to test | locations |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 54ALS00A | NAND GATES | 1 |
| 2 | 54ALS74A | FLIP-FLOP | 1 |
| 3 | IDT71682 | RAM | 2 * ceil ( $v_2$ / m ) |
| 4 | I8206 | EDC Unit | ceil ( $v_1$ / b ) |
| 5 | RNC55H | RESISTOR | 2 |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i = \text{ceil} ( u_i / upp_i )$$

**Explanation of symbols used:**

| | | |
|---|---|---|
| $n_i$ | = | Number of components (physical packages) required for i-th part |
| $u_i$ | = | Number of units (CAD symbols) required for i-th part |
| $upp_i$ | = | Number of units/package for i-th part (from Table 4.0) |
| ceil | = | Round up to nearest integer (e.g. 5.3 is rounded to 6) |
| b | = | Number of data bits for part (from Table 4.0) |
| m | = | Number of memory locations in RAM = $2^a$ |
| a | = | Number of address bits for RAM from Table 4.0 |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |

## BIT TECHNIQUE: ERROR DETECTION AND CORRECTION CODES

## CATEGORY: PENALTY EQUATIONS

|  |  |  |
|---|---|---|
| **AREA (sq.in.)** | = | Sum ( $n_i$ * $a_i$ ) + 15% for traces |
| **WEIGHT (gms)** | = | Sum ( $n_i$ * $w_i$ ) + 10% for solder |
| **POWER (mW)** | = | Sum ( $n_i$ * $p_i$ ) |
| **TEST TIME (ns)** | = | 0 (EDCC is a Concurrent BIT technique) |
| **DELAY (ns)** | = | 67 |

**Explanation of Symbols Used.**

|  |  |  |
|---|---|---|
| Sum | = | Sum over parts in BIT Technique Parts Table (i=1 to 5) |
| $n_i$ | = | Number of component packages for i-th part in Parts Table<br>$n_i$ is calculated according to Component Determination Equations |
| $a_i$ | = | Area in sq.in. of i-th part in Parts Table |
| $p_i$ | = | Power dissipation in mW (typical) for i-th part in Parts Table |
| $w_i$ | = | Weight in grams of i-th part in Parts Table |

## CATEGORY: BIBLIOGRAPHY

1. Len Levine and Ware Meyers, "Semiconducting Memory Reliability with Error Detecting and Correcting Codes," COMPUTER, October 1976, pp 43-50.2.

2. Intel application notes AP-73, "Memory System Reliability with ECC." Intel Memory Components Handbook 1985.

# SCAN DESIGN

# BIT TECHNIQUE DATA PACKAGE

## CATEGORY: SHORT DESCRIPTION / DIAGRAM

Scan BIT techniques provide easy access to buried sequential or combinational circuit elements. Scan design adds hardware overhead, yet has gained widespread popularity due to the following testability attributes it provides:

OBSERVABILITY: The ability to read the state of an entire Line Replaceable Module (LRM) in response to a specified applied test pattern.

CONTROLLABILITY: The ability to initialize a Circuit Under Test (CUT) containing sequential memory elements with more complex test patterns than presets and clears can provide.

PARTITIONING: A SCAN chain becomes a natural partition between logic clusters, thereby promoting the divide and test approach to testing.

By utilizing such circuits as a serial shift register, the SCAN technique is easily accomplished. In bit-serial SCAN, the nodes to be scanned are parallel-shifted into a bit-serial register and then serially shifted out for inspection by the maintenance processor. If the inspection data does not match a good machine state, a Pass/Fail indicator can be set. The Default Design describes one form of the SCAN implementation. Other types of SCAN are:

- Level Sensitive Scan Design (LSSD)
- Random Addressable Scan
- Boundary Scan

Figures 1 and 2 show the Level I and II Block Diagrams for this BIT technique.

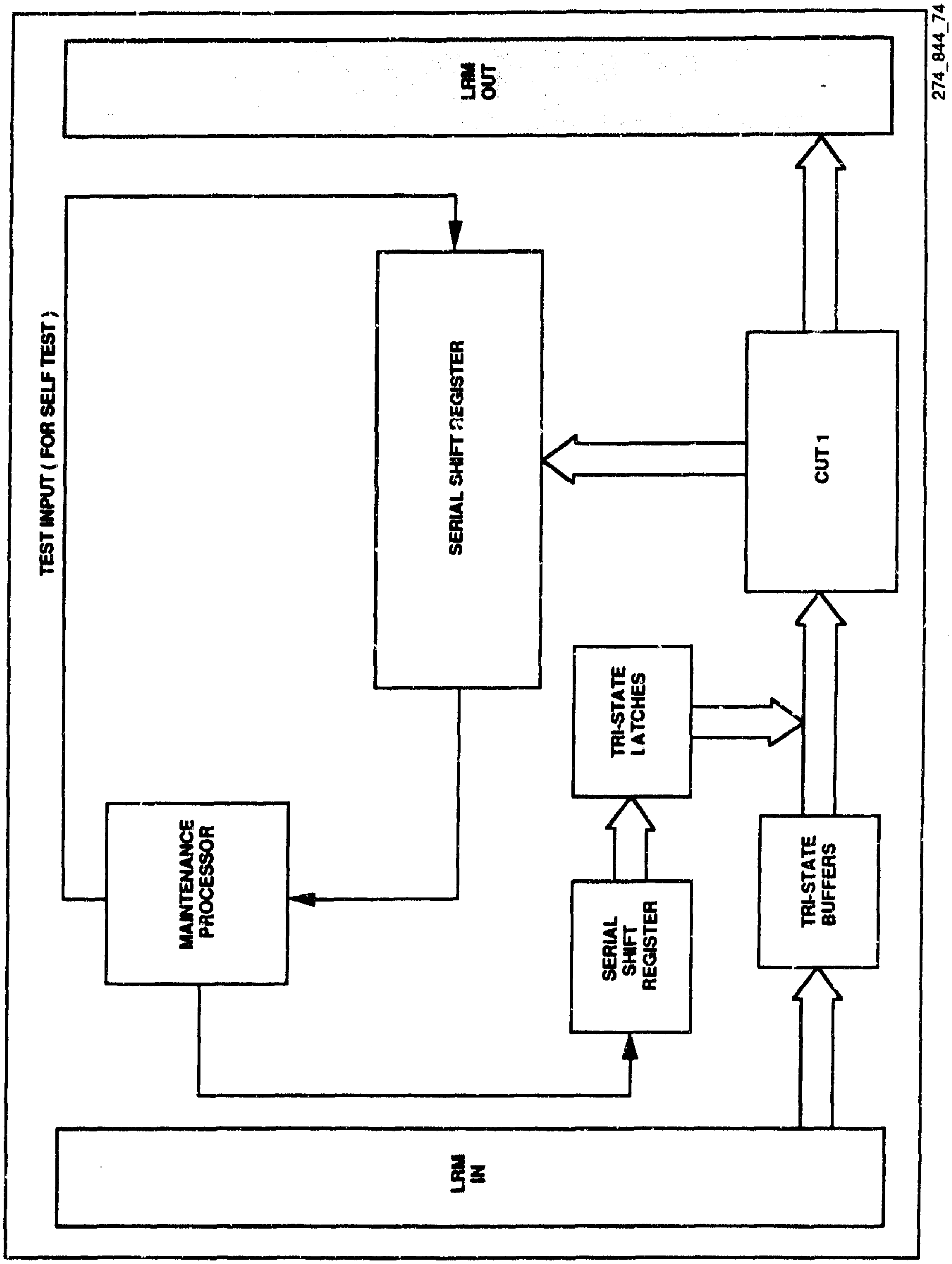


Figure 1 Level I Block Diagram Scan BIT Technique

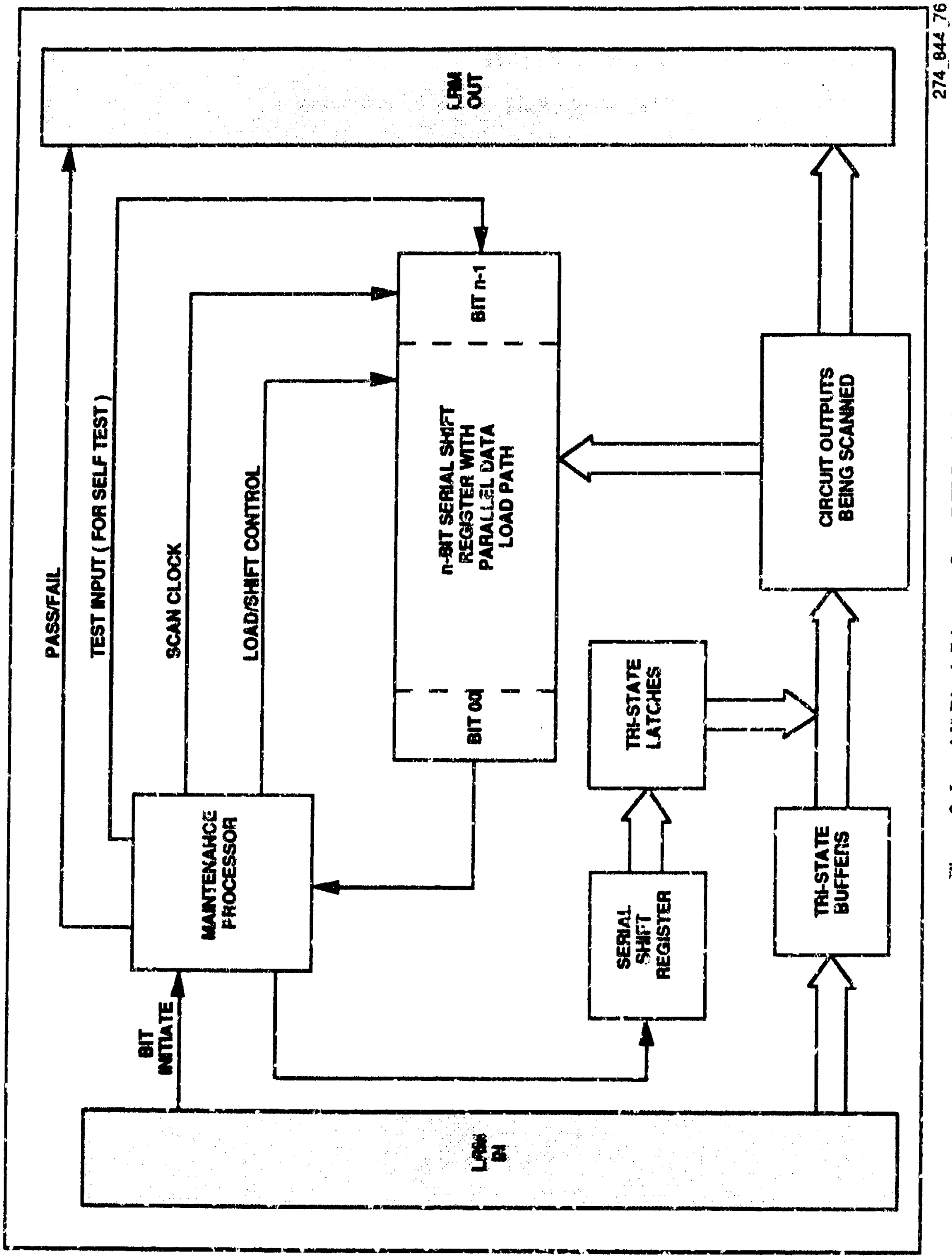


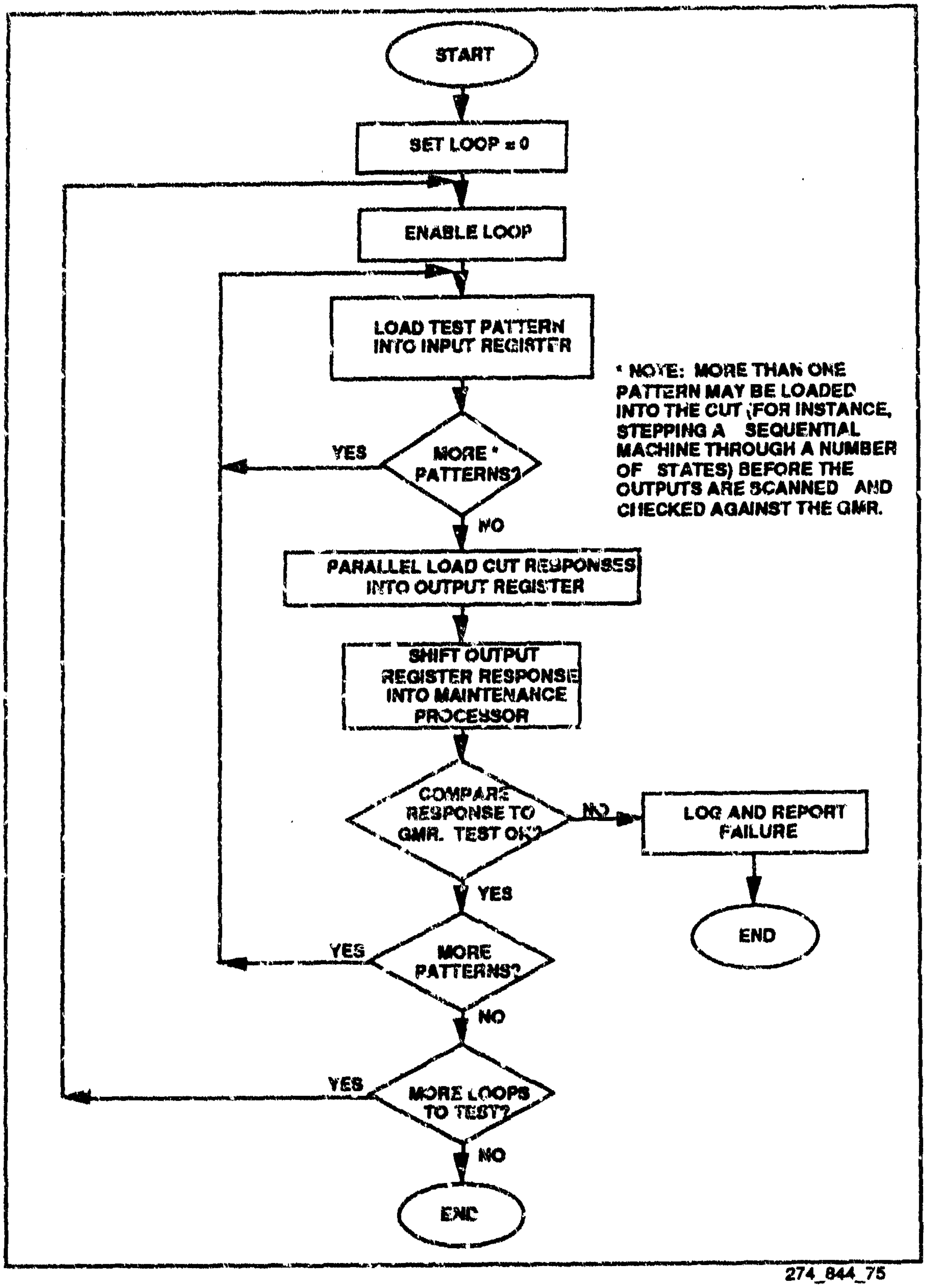
Figure 2 Level II Block Diagram Scan BIT Technique

**BIT TECHNIQUE: SCAN DESIGN**

**CATEGORY: FLOW CHART / DESCRIPTION**

Figure 3 shows the Flow Chart for this BIT technique.

1. Receive BIT INITIATE and set test inputs.
2. Enable first test loop (switches input MUX to test mcde).
3. Shift input test pattern into input register.
4. Load input data into input latch. Input additional sequential patterns if required.
5. Parallel load output data into scan shift registers.
6. Apply READ scan clock and shift out SCAN data.
7. Read SCAN results and compare.
8. If test fails, set Pass/Fail indicator to FAIL and end the test. If test passes, continue on to next test.
9. Input additional sequential patterns if required.
10. Test next loop if required by repeating Steps 2-9. If last loop has been tested, end the test by setting the input MUX to normal mode.


START
SET LOOP = 0
ENABLE LOOP
LOAD TEST PATTERN INTO INPUT REGISTER
MORE * PATTERNS?
YES
NO
* NOTE: MORE THAN ONE PATTERN MAY BE LOADED INTO THE CUT (FOR INSTANCE, STEPPING A SEQUENTIAL MACHINE THROUGH A NUMBER OF STATES) BEFORE THE OUTPUTS ARE SCANNED AND CHECKED AGAINST THE GMR.
PARALLEL LOAD CUT RESPONSES INTO OUTPUT REGISTER
SHIFT OUTPUT REGISTER RESPONSE INTO MAINTENANCE PROCESSOR
COMPARE RESPONSE TO GMR. TEST OK?
NO
LOG AND REPORT FAILURE
END
YES
MORE PATTERNS?
YES
NO
MORE LOOPS TO TEST?
YES
NO
END

274_844_75

Figure 3 BIT Sequence Flow Chart For The Scan Design Technique

## BIT TECHNIQUE: SCAN DESIGN

## CATEGORY: ADVANTAGES

1. Test input and SCAN output latches are completely external to CUT.
2. Allows for parallel load/serial scan output and serial in/parallel load input modes of operation.
3. Only one shift clock is required.
4. Possible to take system snapshots.

## CATEGORY: DISADVANTAGES

1. Serial in and serial out modes require a large amount of test time.
2. Requires control of shift clock by maintenance processor.
3. Requires a maintenance processor for control of the BIT test.
4. Requires a lot of board space and power for large systems.
5. Requires that the normal operation of the CUT be suspended during BIT testing.

## BIT TECHNIQUE: SCAN DESIGN

## CATEGORY: ATTRIBUTES

1. CONCURRENCY

- Non-concurrent. System operation must be suspended to perform the BIT test.

2. TECHNOLOGY

- Typically applicable to CUTs using digital technology.
- BIT circuit is implemented using standard LS and ALS logic ICs, a crystal controlled microprocessor, and an EEPROM.

3. CUT MICROPROCESSOR REQUIRED?

- No.

4. CUT INTERNAL DESIGN REQUIRED?

- No. LRM inputs to CUT must be intercepted and multiplexed with BIT circuit test inputs.

5. AREA PENALTY

- Dependent on the number of CUTs and the number of CUT inputs and outputs. One register stage is needed for each CUT input and output.

6. WEIGHT PENALTY

- Roughly proportional to the area penalty. Higher if CUTs have large numbers of outputs to be monitored or large numbers of inputs, or both.

7. POWER PENALTY

- Power penalty will depend on the number of SCAN registers (which depends on the number of CUTs and CUT inputs and outputs) and the microprocessor power dissipation.

**BIT TECHNIQUE: SCAN DESIGN**

**CATEGORY: ATTRIBUTES, Contd**

8. TIMING PENALTY

- Throughput Delay – In normal operation, there will be a slight increase in throughput delay due to the addition of tri-state buffers to the input path.
- Test Time – During BIT testing, normal operation of the CUT is suspended. BIT test time is long due to the serial transfer of data.

9. RELIABILITY

- Slight decrease in reliability due to the addition of a number of register chips and a maintenance processor. The number of register chips is proportional to the number of CUTs and CUT inputs and outputs.

10. CONCEPTUAL COMPLEXITY

- Moderate complexity due to the presence of a maintenance processor.

11. HARDWARE/SOFTWARE/FIRMWARE

- Hardware is used to implement the input and output registers, maintenance processor, test pattern memory, and control functions.
- Software in the maintenance processor controls the loading of input test patterns, the loading of output data, and the comparison of output data.

12. DESIGN COST

- Hardware cost is minimal with the use of off-the-shelf chips. Additional non-recurring cost is incurred for the programming of the maintenance processor.

**BIT TECHNIQUE: SCAN DESIGN**

**CATEGORY: ATTRIBUTES, Contd**

13. MEMORY REQUIREMENTS

- Test patterns and Good Machine Responses are stored in EEPROM.
- The maintenance processor has built-in EPROM and RAM to store program code and data.

14. BIT CIRCUITRY SELF-TESTABLE?

- The BIT circuit is fully self-testable. The maintenance processor can be used to load and scan both the input test pattern and the SCAN output registers for proper operation. Also, test software can be written to check the function of the main program.

15. STAND-ALONE (SELF-CONTAINED BIT)?

- Yes.

16. NOTES

- None.

## BIT TECHNIQUE: SCAN DESIGN

## CATEGORY: DEFAULT DESIGN

Refer to Figure 4 for the Default Design schematic.

The SCAN BIT technique implemented in the Default Design provides hardware and software to support the testing of up to 4 CUTs. The circuitry for only 1 CUT is shown in the interest of clarity (see the cross-hatched area of Figure 4). This design incorporates the default values for the number of CUT inputs (8) and CUT outputs (16) to be tested. These can easily be tailored to the particular CUT by adding or deleting hardware.

Functionally, the SCAN technique is implemented using a maintenance processor to control the test, EEPROM memory to store test patterns, an input shift register with parallel output to load the test patterns into the CUT, a MUX to select between CUT normal inputs and test pattern inputs, an output shift register with parallel load to accept CUT test outputs and shift them into the maintenance processor, and associated glue logic.

The maintenance processor selected for the design was the Intel I87C51 8-bit Microcontroller. It features 4 configurable 8-bit ports, Port 0 through Port 3. Port 0 is multiplexed to put out the low-order address byte and to receive data during accesses to the EEPROM memory. Port 1 is configured as a general purpose test bus. It can be used to communicate the results of the BIT test to the system processor which can then be used to fault isolate errors to the component level. Port 2 provides three address lines to the EEPROM as well as a number of signals to control the BIT test. It also outputs the _PASS/FAIL signal (Port 2, bit 7 or P2.7 for short) to the system. Port 3 generates the read and write control lines to the EEPROM, accepts the BIT_INITIATE (P3.2) interrupt from the system, outputs the serial data to the input shift register, accepts input data from the output shift register, generates the clock to control the shift registers, and provides the control signals to generate the loop enables. The control signals are generated by software stored in the on-chip EPROM. Also, the software performs the comparison of the output data with the Good Machine Response (GMR) and outputs the results through the _PASS/FAIL signal.

U2, a 54ALS373 Octal Latch, stores the low-order address byte as is typical for processors with multiplexed address/data busses.

U3 is a 2817A 2K X8 EEPROM used to store test patterns for the BIT test and, optionally, Good Machine Responses.

U4 is a 54ALS138 3-line to 8-line decoder and enables one of 4 CUT BITs to be tested. It uses three outputs from the maintenance processor, P3.3, P3.4 and P3.5, to enable BIT testing and to select which channel is tested.

**BIT TECHNIQUE: SCAN DESIGN**

**CATEGORY: DEFAULT DESIGN, Contd**

The active _LOOP_EN output for a particular channel selects the test pattern inputs, disables the normal CUT inputs (U9 and U10), and enables the output shift registers (U11 and U12).
U5 is a 54ALS253 1 of 4 line data selector and is used to select which CUT's data output from the output shift register is routed to the maintenance processor data input (P3.0).

U6 (54ALS32) contains OR gates used to enable or disable the clock output (P3.1) depending the R/_W control output (P2.4) from the maintenance processor. U6A enables the RD_CLK lines when the R/_W is high while U6B enables the WR_CLK lines when R/_W is low.

U8 is a buffer/line driver for the DATA_OUT, RD_CLK, WR_CLK and LD lines coming out of the maintenance processor and the control logic.

U9 is a 54LS595 serial input shift register with parallel tri-state output latches. It is used to shift in test pattern data supplied by the maintenance processor (P3.0) through U8A. It uses the _LOOP_EN signal from U4 to enable the latch outputs, the LD signal from the maintenance processor (P2.3) through U8B to load the parallel output, and the WR_CLK from U8B as a clock to shift in the data. Its outputs are bussed with the output of U1O, a 54ALS244A tri-state buffer whose inputs are connected to the LRM outputs which would normally connect to the CUT inputs. The LOOP_EN signal is used to control which tri-state outputs are enabled. In this way the CUT inputs are selected to be either the test pattern inputs or the normal inputs from the LRM. In normal operation, the U10 buffers are in-line with the inputs to the CUT adding a propagation delay of 18 nsec maximum. The circuit can easily be expanded to accommodate more inputs by cascading 54LS595 shift registers and adding more 54ALS244A tri-state buffers. To cascade 54LS595 chips connect the $Q_{HP}$ output (pin 9) to the serial input (pin 14) of the successive chip.

U11 and U12 are 54ALS165A 8-bit parallel in, serial out shift registers. The CUT output nodes being tested are parallel loaded into the register and then serially shifted out to the maintenance processor (P3.0) through U5. The circuit can easily be expanded to accommodate more outputs by adding more chips. The $Q_H$ output (pin 9) of each chip connects to the CLK input (pin 2) of the successive chip. Control signals are the RD_CLK (from U8A) used to shift the data out of the register, LOOP_EN (from U4) to enable the clock input to the shift register, and LD (from U8B) to control whether the register is in the shift or load parallel data mode.

**BIT TECHNIQUE: SCAN DESIGN**

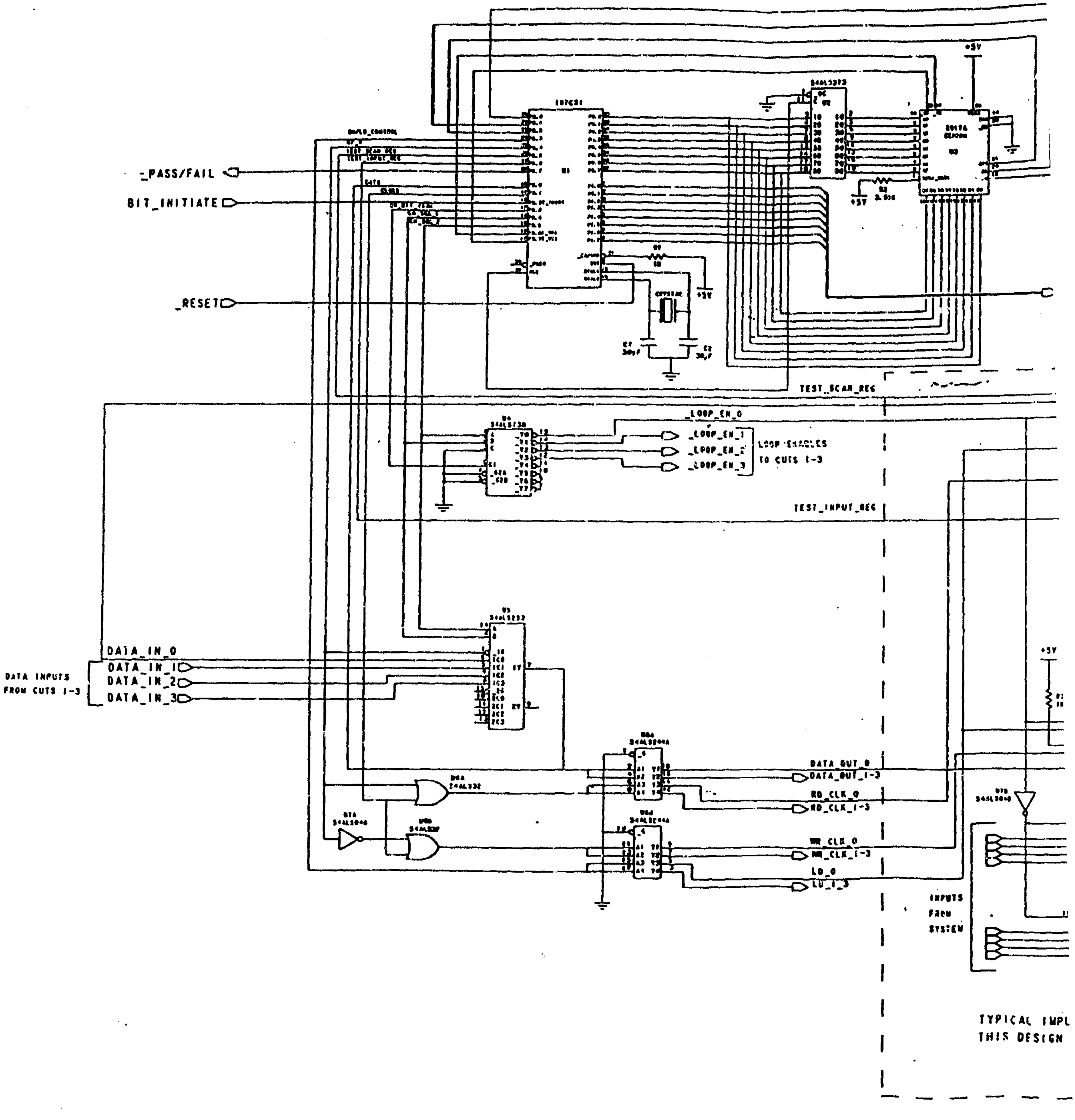
**CATEGORY: DEFAULT DESIGN, Contd**

Two test signals are provided by the maintenance processor (P2.5 and P2.6) to test the input and output registers. TEST_SCAN_REG inputs serial data from the processor into the output register. Data is shifted out in the usual fashion and checked against the input for errors. For the other test signal, the input register is loaded with a test pattern in the usual manner and data is shifted out to the maintenance processor through the TEST_INPUT_REG line for comparison. The software is also exercised completing the self-test of the BIT circuitry.

Test Philosophy: Any combinational circuit can be tested by loading the CUT inputs with a test pattern and checking the CUT output to a GMR. Sequential machines can also be tested by using the clock signal for the machine as one of the inputs to the CUT. The processor can be programmed to manipulate the clock and the CUT outputs can be tested for any input condition in any machine state.

Certain limitations do apply to this technique due to its slow sped. For instance, outputs of one-shots cannot be tested if their pulse widths are short (< 100 msec).

Conclusion: The SCAN BIT technique can be useful to many applications using sequential or combinational logic if normal operation of the CUT can be suspended during the BIT test, the slow speed of the test can be tolerated, and the large hardware overhead is acceptable.



87C51
U1
54ALS373
U2
+5V
U3
R2
+5V
R1
CRYSTAL
C1
30pF
C2
30pF
+5V
_PASS/FAIL
BIT_INITIATE
_RESET
TEST_SCAN_REG
TEST_INPUT_REG
U4
54ALS138
_LOOP_EN_0
_LOOP_EN_1
_LOOP_EN_2
_LOOP_EN_3
LOOP ENABLES
TO CUTS 1-3
U5
54ALS253
DATA INPUTS
FROM CUTS 1-3
DATA_IN_0
DATA_IN_1
DATA_IN_2
DATA_IN_3
U6A
54ALS244A
U6A
54ALS32
U7A
54ALS04
U6B
54ALS32
U6A
54ALS244A
DATA_OUT_0
DATA_OUT_1-3
RD_CLK_0
RD_CLK_1-3
WR_CLK_0
WR_CLK_1-3
LD_0
LD_1_3
+5V
U7B
54ALS04
INPUTS
FROM
SYSTEM
TYPICAL IMPL
THIS DESIGN

1

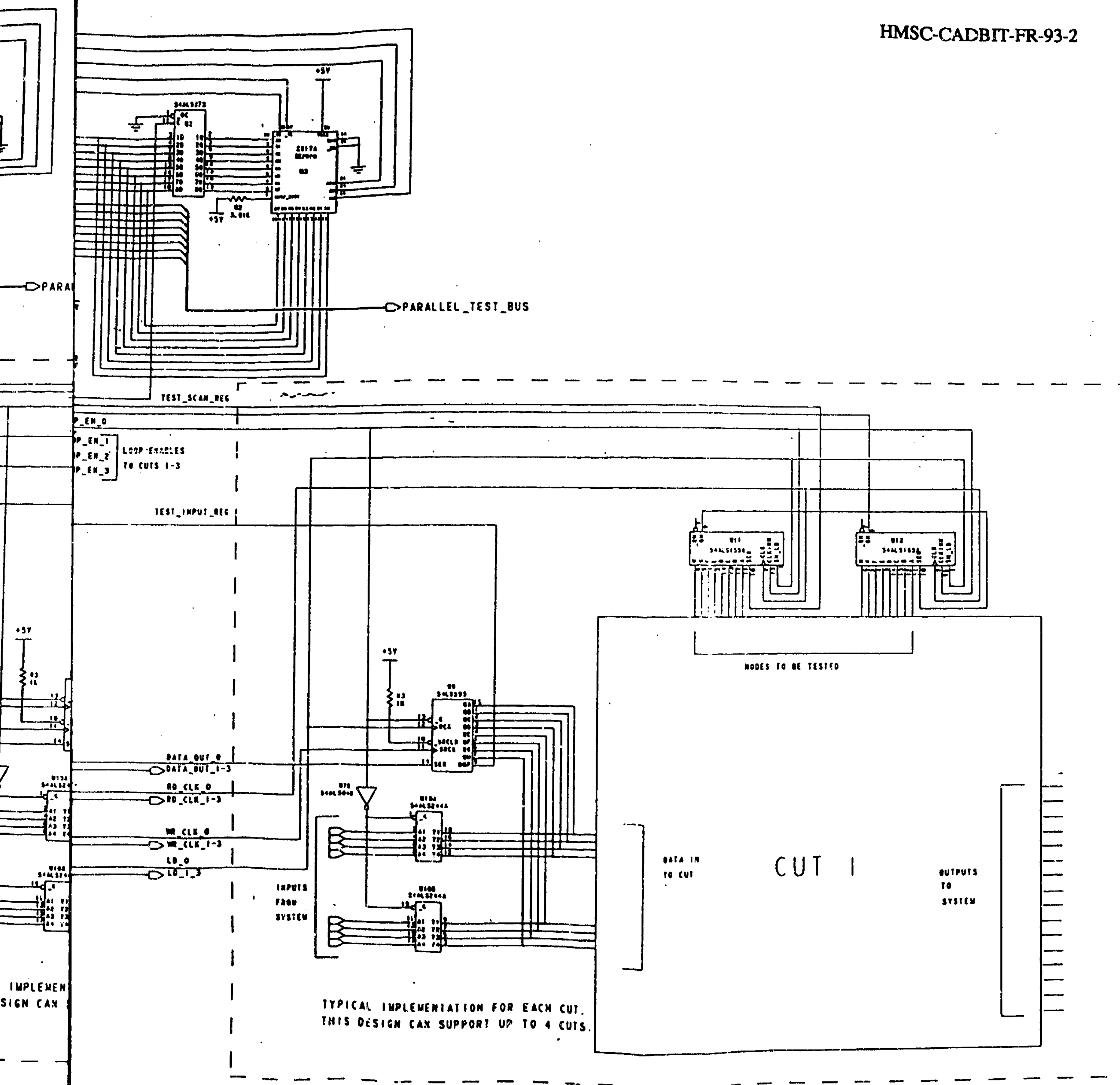


Figure 4 Scan Design – Default Design

## BIT TECHNIQUE: SCAN DESIGN

## CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM

Figure 5 shows the BIT Technique Insertion Diagram for this BIT technique.

BIT ELEMENT 1

1. Connect the BIT_INITIATE, _RESET, and _PASS/FAIL signals between the maintenance processor of BIT ELEMENT 1 (BE1) and the LRM.

2. Connect the PARALLEL_TEST_BUS outputs of BE1 to the LRM.

BIT ELEMENT 2

1. For CUT 1, connect WR_CLK_0 output of BE1 to BIT ELEMENT 2 (BE2). If there are more CUTs (up to 4 total), connect WR_CLK_1-3 to the BE2s for those CUTs.

2. Connect the DATA_OUT_0 from BE1 to BE2 of CUT 1. If there are more CUTs (up to 4 total), connect DATA_OUT_1-3 to the BE2s of the other CUTs.

3. Connect the normal CUT 1 inputs from the LRM to the BE2 associated with CUT 1. If there are more CUTs (up to 4 total), connect the normal CUT inputs to their associated BE2s.

4. Connect the multiplexed outputs of BE2 to their associated CUT inputs.

5. Connect the TEST_INPUT_REG signal between BE1 and BE2.

BIT ELEMENT 3

1. For CUT 1, connect the RD_CLK_0 output from BE1 to BIT ELEMENT 3 (BE3). If there are more CUTs (up to 4 total), connect RD_CLK_1-3 to BE3 for those CUTs.

2. Connect the LD_0 AND _LOOP_EN_0 to both BE2 and BE3 associated with CUT 1. If there are more CUTs (up to 4 total), connect LD_1-3 to the BE2s and BE3s of the remaining CUTs and _LOOP_EN_1 to BE2 and BE3 of CUT2, _LOOP_EN_2 to BE2 and BE3 of CUT 3, and _LOOP_EN_3 to BE2 and BE3 or CUT 4.

3. Connect the DATA_IN_0 from BE3 TO BE1. If there are more CUTs (up to 4 total), connect DATA_IN_1 from BE3 to BE1 of CUT 2, DATA_IN_2 from BE3 to BE1 of CUT 3, and DATA_IN_3 from BE3 to BE1 of CUT 4.

BIT ELEMENT 3

4. Connect all CUT test outputs to their corresponding BE3 parallel data inputs.

5. Connect the TEST_SCAN_REG between BE3 and BE1.

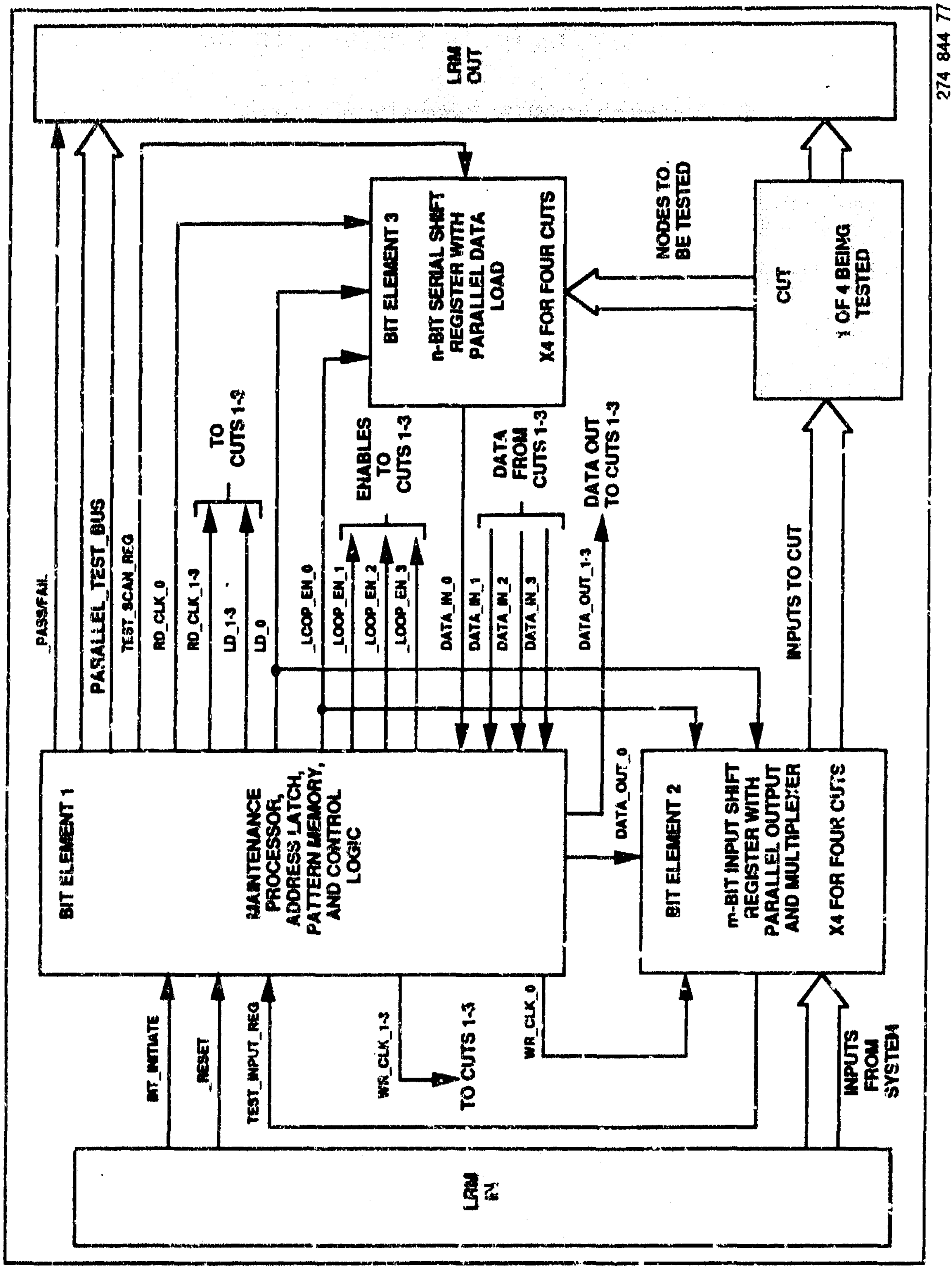


Figure 5 BIT Insertion Block Diagram Scan BIT Technique

## BIT TECHNIQUE: SCAN DESIGN

## CATEGORY: PENALTY EQUATIONS

**AREA (sq.in.)** = Sum ( $n_i$ * $a_i$ ) + 15% for traces

**WEIGHT (gms)** = Sum ( $n_i$ * $w_i$ ) + 10% for solder

**POWER (mW)** = Sum ( $n_i$ * $p_i$ )

**TEST TIME (ns)** =

**DELAY (ns)** =

**Explanation of Symbols Used.**

| | | |
|---|---|---|
| Sum | = | Sum over parts in BIT Technique Parts Table (i=1 to 11) |
| $n_i$ | = | Number of component packages for i-th part in Parts Table<br>$n_i$ is calculated according to Component Determination Equations |
| $a_i$ | = | Area in sq.in. of i-th part in Parts Table |
| $p_i$ | = | Power dissipation in mW (typical) for i-th part in Parts Table |
| $w_i$ | = | Weight in grams of i-th part in Parts Table |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |

## CATEGORY: BIBLIOGRAPHY

1. E. J. McCluskey, "A Survey of Design for Testability Scan Techniques", *VLSI Design*, December 1984

2. B. Eichelberger and T. W. Williams. "A Logic Design Structure for LSI Testability", Fourteenth Annual Design Automation Conference, New Orleans, June 1977, pp. 462-467

3. M. J. Y. Williams and J. B. Angell, "Enhancing Testability of Large Scale Integrated Circuits Via Test Points and Additional Logic", IEEE vol no. 1 (Jan 1978) pp.46-60

# DIGITAL WRAPAROUND
# BIT TECHNIQUE DATA PACKAGE

## CATEGORY: SHORT DESCRIPTION

Digital Wraparound is a non concurrent Built-In-Test (BIT) technique. This technique consists of hardware and software (firmware in Read-Only Memory (ROM)) and specifically requires a microprocessor, some digital output devices and some digital input devices on board as part of the Circuit Under Test (CUT).

The technique consists of adding the necessary circuitry so that upon BIT INITIATE, the digital data leaving the digital output devices can be routed to the digital input device on the Line Replaceable Module (LRM) by adding digital gates to wrap the outputs around to the inputs. An appropriate BIT routine is stored in ROM along with test data to control the data transfer and compare the data received with the data transmitted. A mismatch will indicate a failure.

The Microprocessor BIT technique (a related digital BIT technique), checks out the internal components of the microprocessor system. The wraparound BIT can be used to extend the microprocessor BIT to include the I/O.

Figures 1 and 2 show the Level I and II Block Diagrams for this BIT technique.

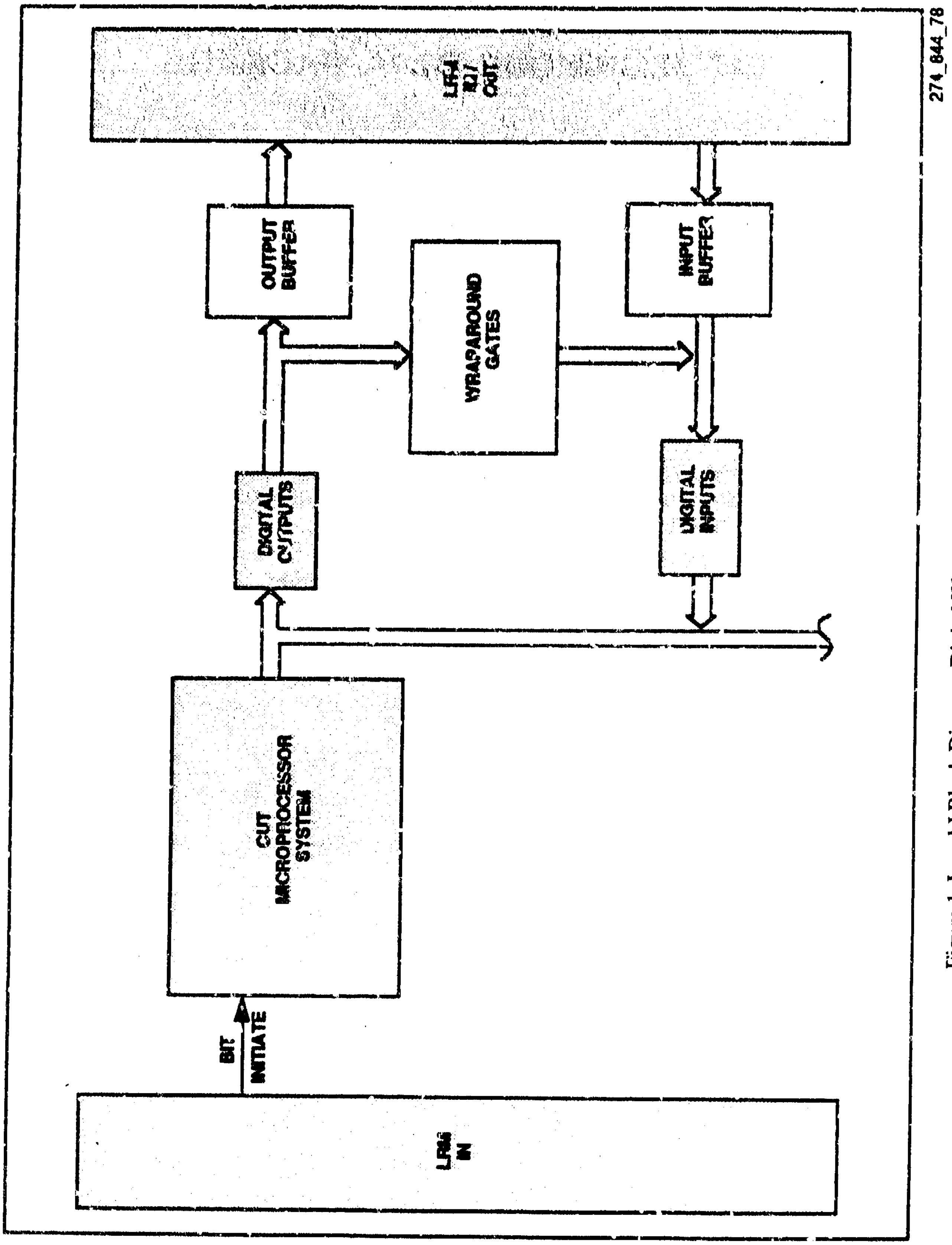


Figure 1 Level I Block Diagram Digital Wraparound As A BIT Technique

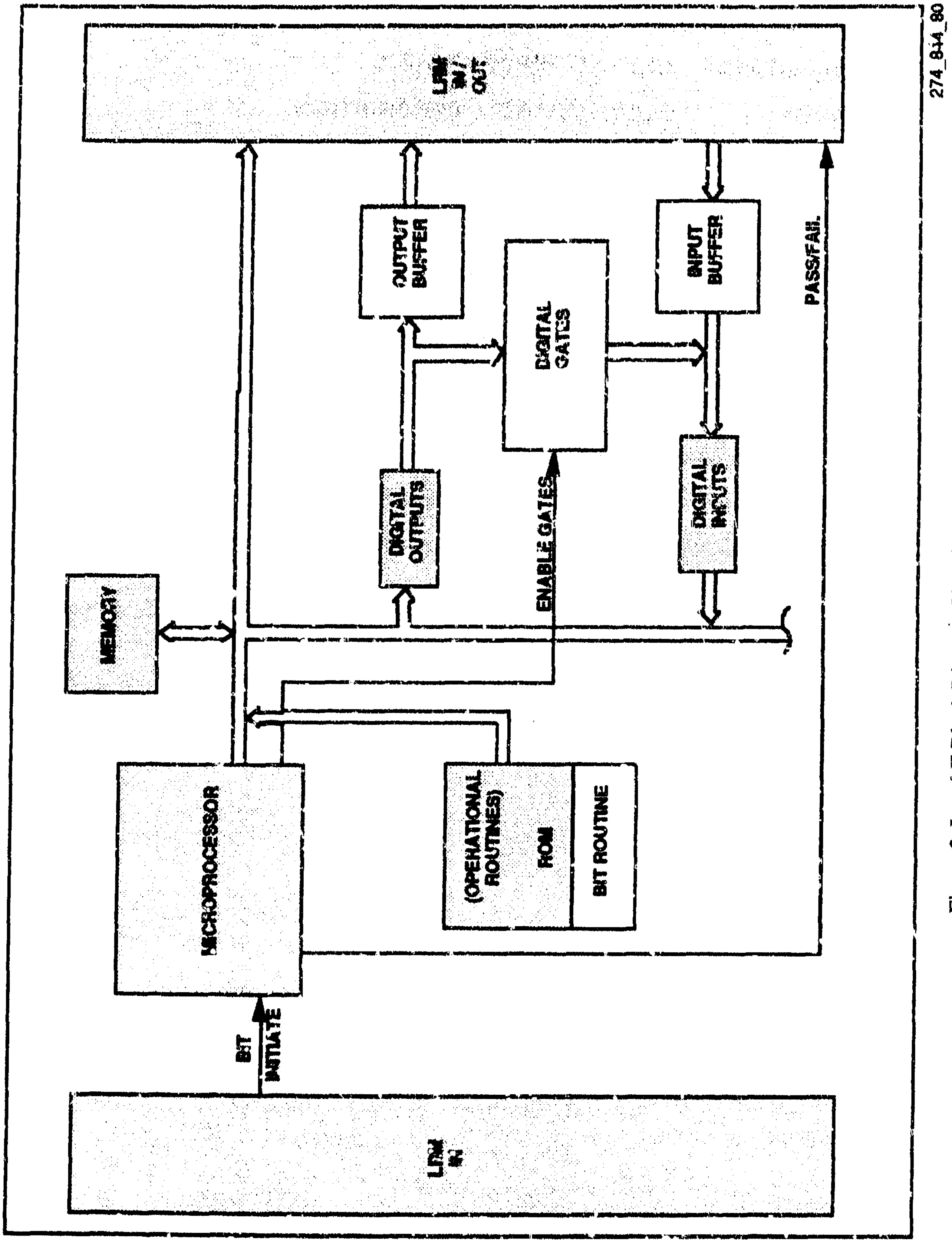


Figure 2 Level II Block Diagram Utilizing Digital Wraparound

**BIT TECHNIQUE: DIGITAL WRAPAROUND**

**CATEGORY: FLOW CHART / DESCRIPTION**

Figure 3 shows the Flow Chart for this BIT technique.

1. A 'BIT INITIATE' signal is input to the LRM, so testing can begin.

2. Initialize the Circuit Under Test and set the Pass/Fail Flip-Flop to PASS.

3. Before applying a signal, enable the wraparound gates that are going to be used for that particular test and disable system input and output buffers.

4. Apply the ROM test pattern to the output device(s).

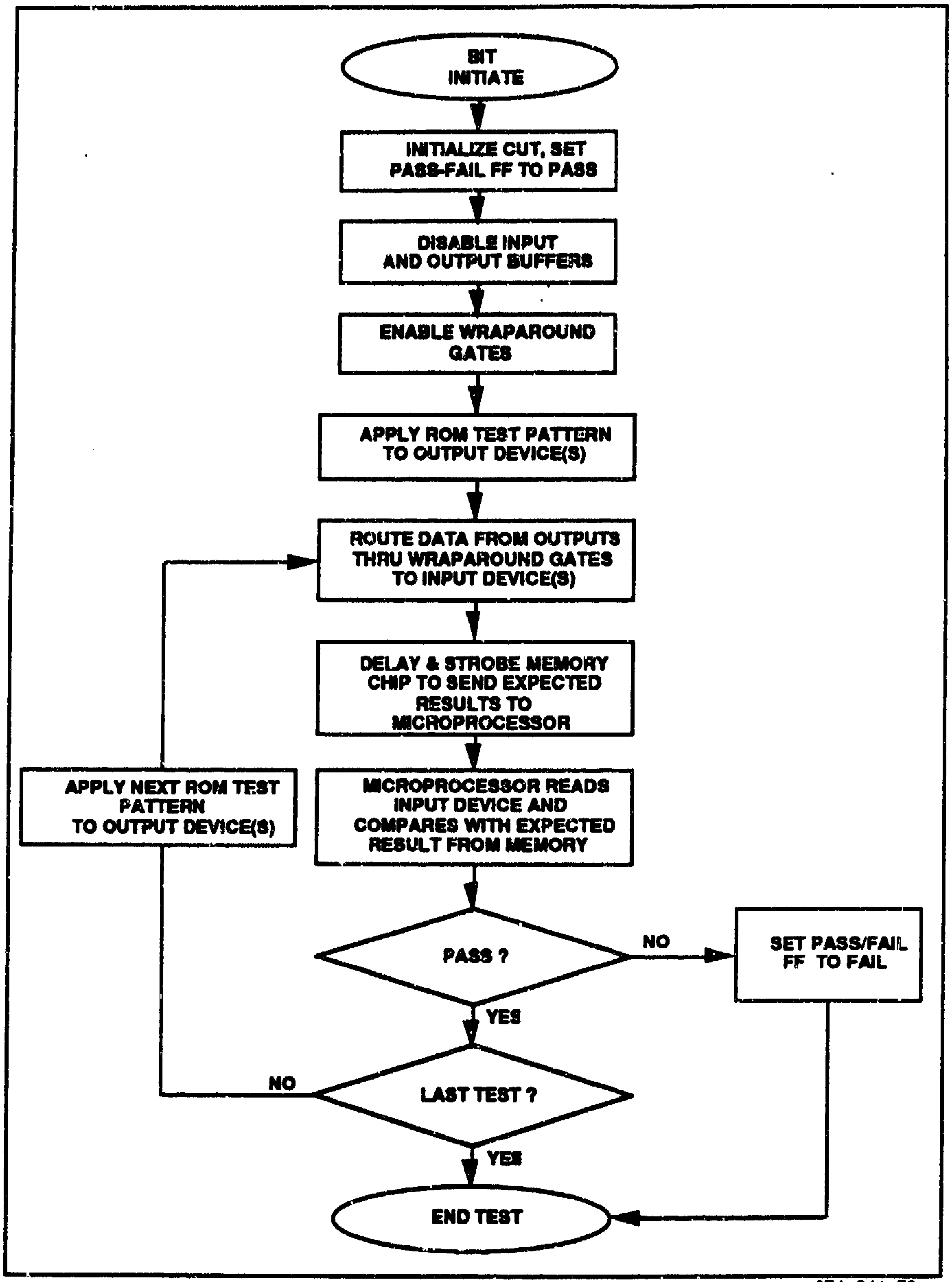
5. At this point, the test data is routed from the outputs through the proper enabled wraparound gates and into the input device(s).

6. Delay and strobe the memory chip to send the expected results to the microprocessor.

7. Microprocessor reads the results from the input devices(s) and compares it with expected result from memory.

8. If comparison fails, set Pass/Fail Flip-Flop to FAIL and end test. If comparison passes, continue to next step.

9. If this is the last test pattern, then end the test. Otherwise, go back to STEP 4 and continue.


BIT
INITIATE
INITIALIZE CUT, SET PASS-FAIL FF TO PASS
DISABLE INPUT AND OUTPUT BUFFERS
ENABLE WRAPAROUND GATES
APPLY ROM TEST PATTERN TO OUTPUT DEVICE(S)
ROUTE DATA FROM OUTPUTS THRU WRAPAROUND GATES TO INPUT DEVICE(S)
DELAY & STROBE MEMORY CHIP TO SEND EXPECTED RESULTS TO MICROPROCESSOR
APPLY NEXT ROM TEST PATTERN TO OUTPUT DEVICE(S)
MICROPROCESSOR READS INPUT DEVICE AND COMPARES WITH EXPECTED RESULT FROM MEMORY
PASS ?
NO
SET PASS/FAIL FF TO FAIL
YES
NO
LAST TEST ?
YES
END TEST

274_844_79

Figure 3 BIT Sequence Flow Chart For Digital Wraparound

**BIT TECHNIQUE: DIGITAL WRAPAROUND**

**CATEGORY: ADVANTAGES**

1. Only requires minimal hardware and is a conceptually simple design which is easy to implement.

2. Chips that are needed are readily available (the wraparound device is generally standard gates of the same logic family used in the digital I/O.)

3. This technique may also be used in conjunction with "MICROPROCESSOR BIT", another Computer-Aided Design for Built-In-Test (CADBIT) technique, to extend the BIT coverage to include the I/O chips (which are not normally checked out with the Microprocessor BIT technique.)

**BIT TECHNIQUE: DIGITAL WRAPAROUND**

**CATEGORY: DISADVANTAGES**

1. This technique only checks out a small portion of the LRM.

2. If the number of test patterns needed to completely test the digital I/O is large (for example, a MIL-STD-1553B interface), then additional ROMs may have to be added to store the test patterns. This will increase the real estate penalty. However, if the I/O devices are simply buffers, only a few patterns will be required, and most ROMs will have spare locations.

**BIT TECHNIQUE: DIGITAL WRAPAROUND**

**CATEGORY: ATTRIBUTES**

1. CONCURRENCY

- Non-concurrent

2. TECHNOLOGY

- Digital

3. CUT MICROPROCESSOR REQUIRED?

- Yes

4. CUT INTERNAL DESIGN REQUIRED?

- No

5. REAL ESTATE PENALTY

- Small – Only requires several integrated circuit packages of gates.
- ROMs – Depends on number of test patterns (as patterns increase, any spare ROM locations may be depleted, therefore an additional ROM(s) may have to be added).
- If number of I/O lines is large, there will be a corresponding increase in number of wraparound gates required.

6. WEIGHT PENALTY

- Small – Proportional to real estate penalty

7. POWER PENALTY

- Small – Just requires additional power to wraparound gates and additional ROM if needed.

8. RELIABILITY IMPACT

- Minimal impact since only a few gates of the same logic family as the I/O devices are added.

## BIT TECHNIQUE: DIGITAL WRAPAROUND

## CATEGORY: ATTRIBUTES, Contd

9. TIMING PENALTY

- Throughput delay – equal to propagation delay of I/O buffer.
- Test time – sum of test initiation time and number of test patterns multiplied by the pattern application rate.

10. CONCEPTUAL COMPLEXITY

- Straightforward

11. HARDWARE/SOFTWARE/COMBO

- Hardware present/test patterns in firmware.

12. STAND ALONE?

- Yes

13. IS BITE SELF-TESTABLE?

- No.

14. DESIGN COST

- Minimized since wraparound chips needed are readily available.
- The CUT microprocessor can also be used for other BIT techniques.
- Engineering time to create patterns depends on complexity of I/O chips to be tested.

15. MEMORY REQUIREMENTS

- ROM needed to store test patterns

16. NOTES

- None

**BIT TECHNIQUE: DIGITAL WRAPAROUND**

**CATEGORY: DEFAULT DESIGN**

This section describes the sequence of events for the DIGITAL WRAPAROUND BIT technique default design. Refer to Figure 4 for the Default Design schematic.

1. When the microcontroller receives a TEST_INITIATE signal to its port P1.4, the test begins.

2. Output from P1.6 enables wraparound gates U8A and U8B. The inverted P1.6 signal disables inputs and outputs octal buffers U7A, U7B, U9A, and U9B.

3. Set the Pass/Fail indicator P1.5 to PASS.

4. Apply test pattern to U4 as the output device. Routes the pattern back to the input device through U8A and U8B as the wraparound gates. The microcontroller captures the response through the input device U5 and compares it with expected results from memory internally.

5. Step 4 is repeated until all test patterns pass. If any comparison fails, set Pass/Fail indicator P1.5 to FAIL and end test.

6. When test is finished, output from P1.6 disables wraparound gates U8A and U8B. The inverted P1.6 signal enables inputs and outputs octal buffers U7A, U7B, U9A, and U9B. The circuit resumes normal operation.

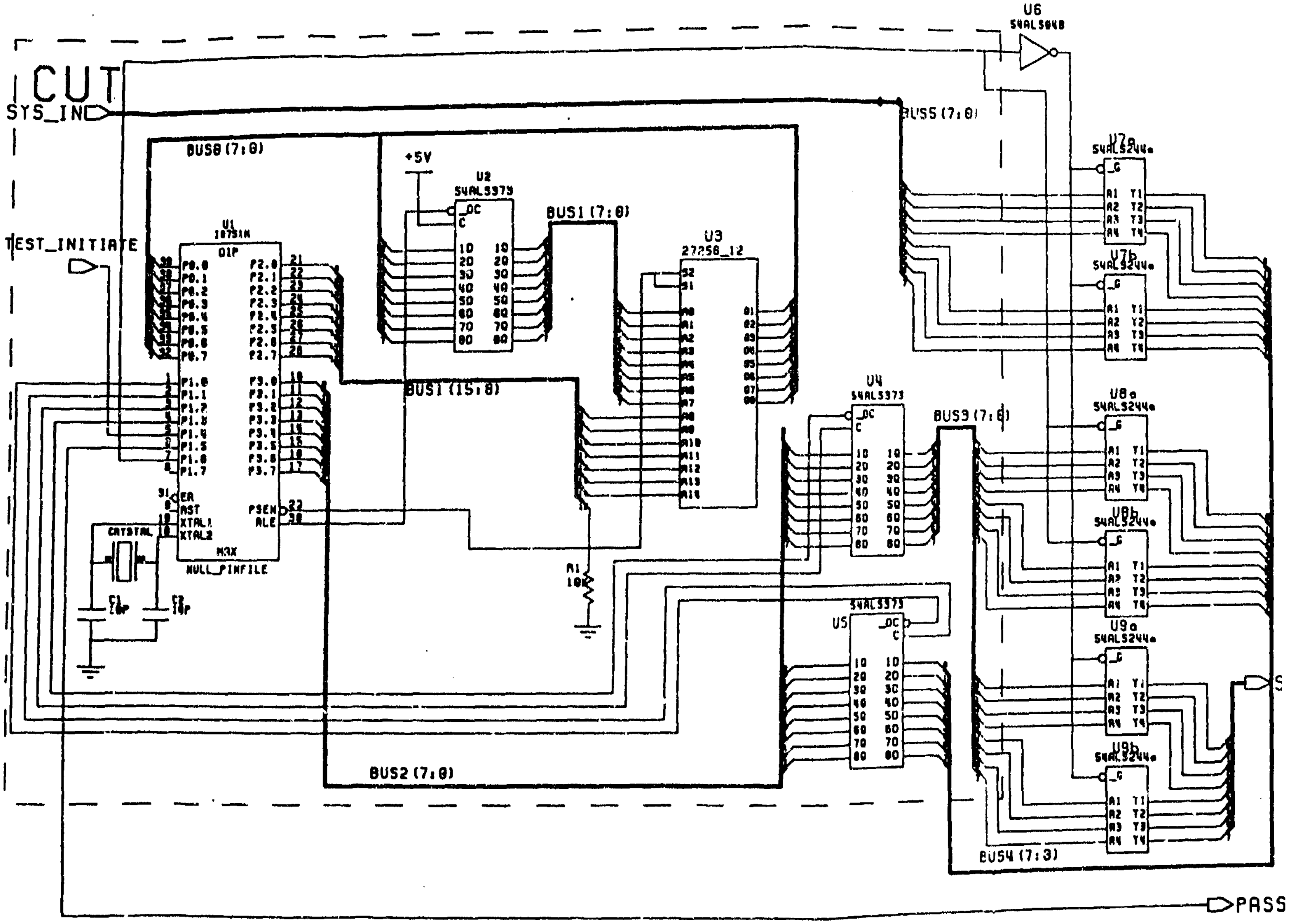


Figure 4  Digital Wraparound – Default Design

1

2

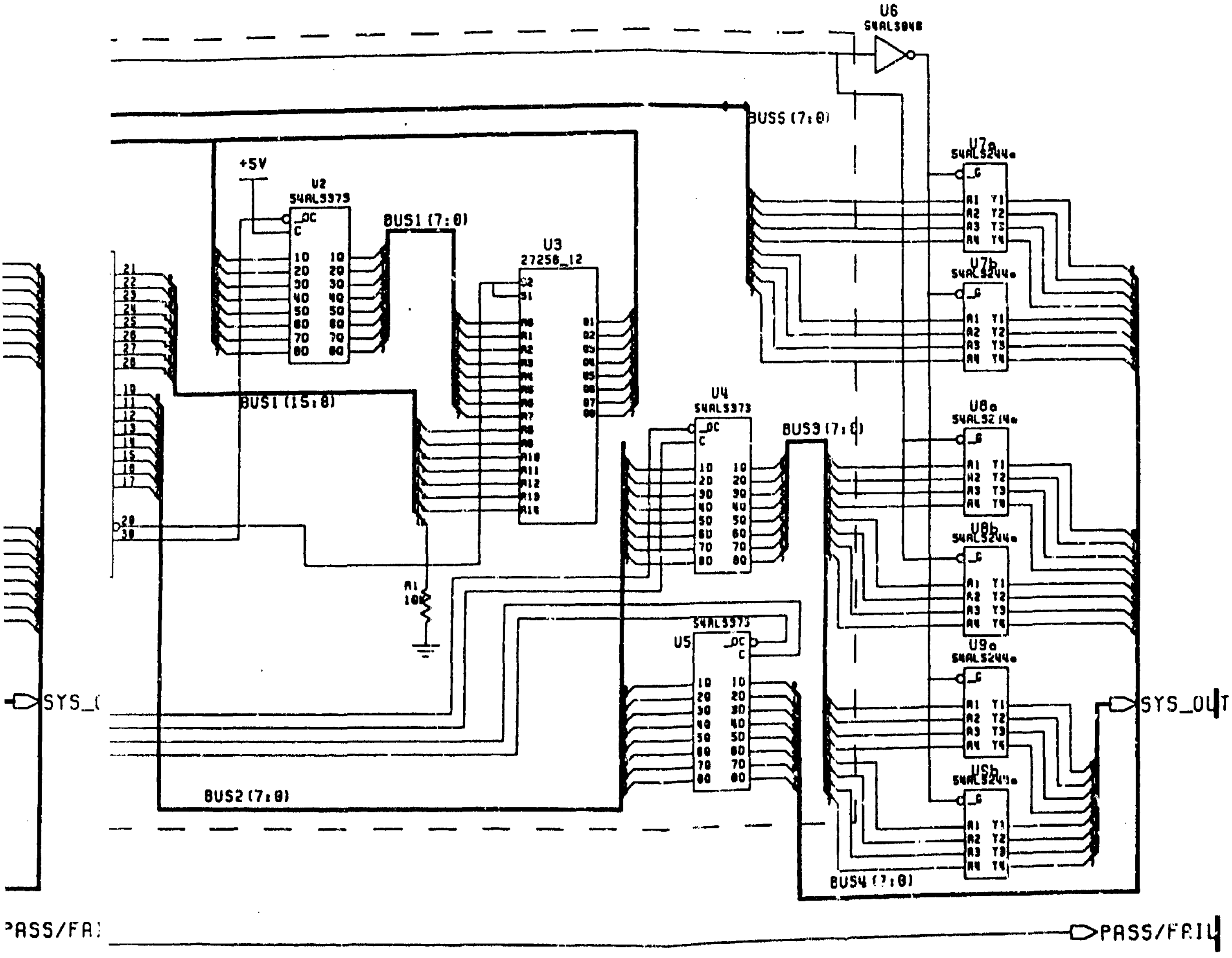


Figure 4 Digital Wraparound – Default Design

**BIT TECHNIQUE: DIGITAL WRAPAROUND**

**CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM**

Figure 5 shows the BIT Technique Block Diagram for this BIT technique.

Connections instructions are shown in Figures 6-9 as follows:

Figure 6 – Connection instructions for BTID Block 1 – Output Buffers

Figure 7 – Connection instructions for BTID Block 2 – Digital Gates

Figure 8 – Connection instructions for BTID Block 3 – Input Buffers

Figure 9 – Connection instructions for Pass/Fail Flip-Flop

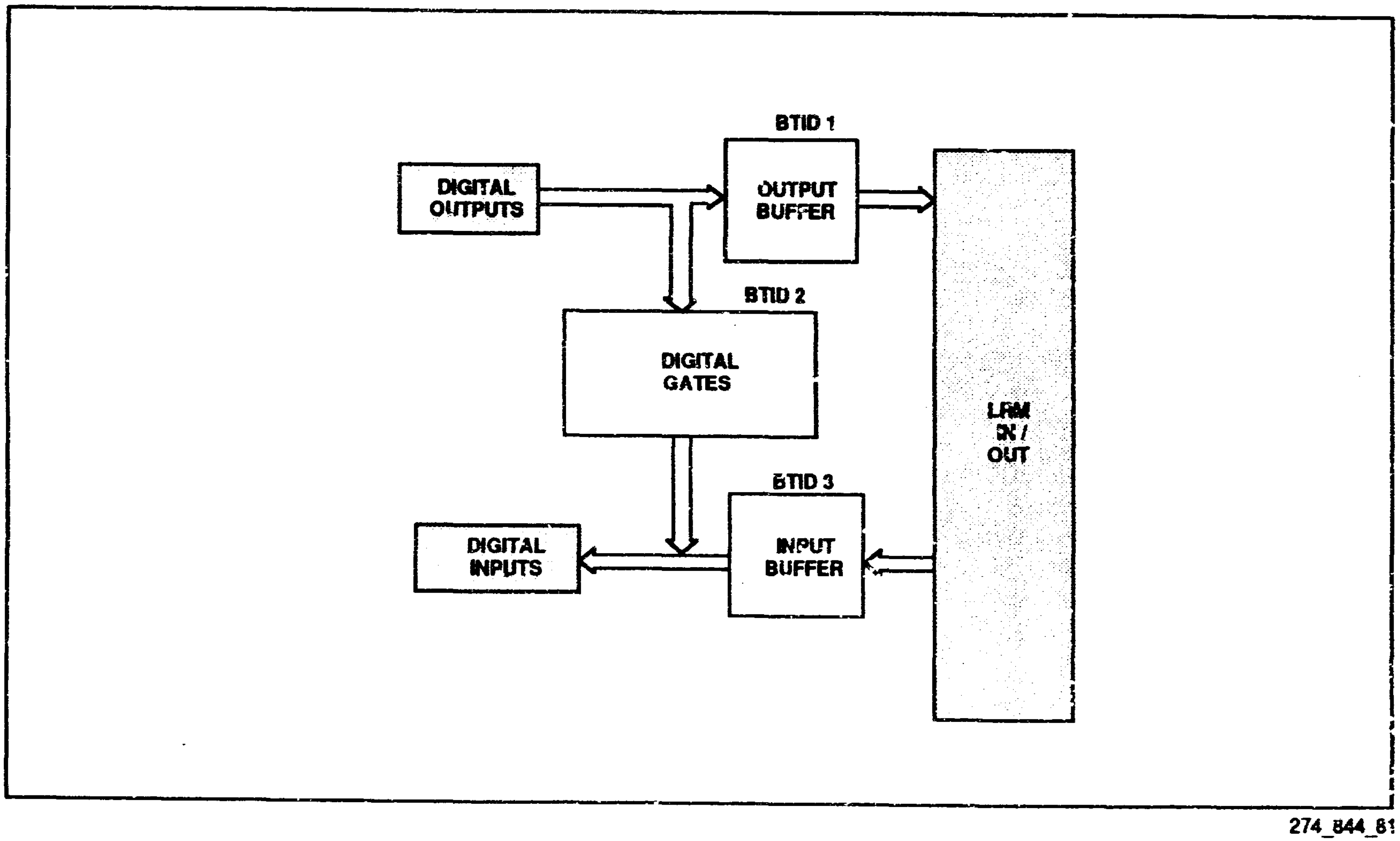
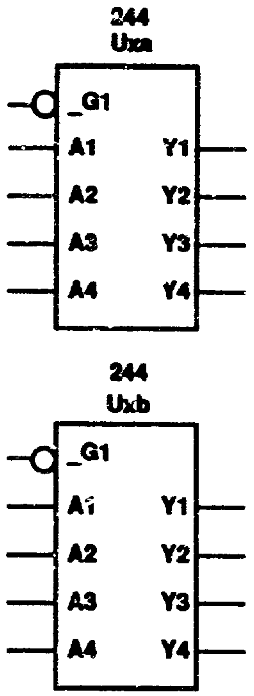


Figure 5 Digitai Wraparound BTID - Top Level

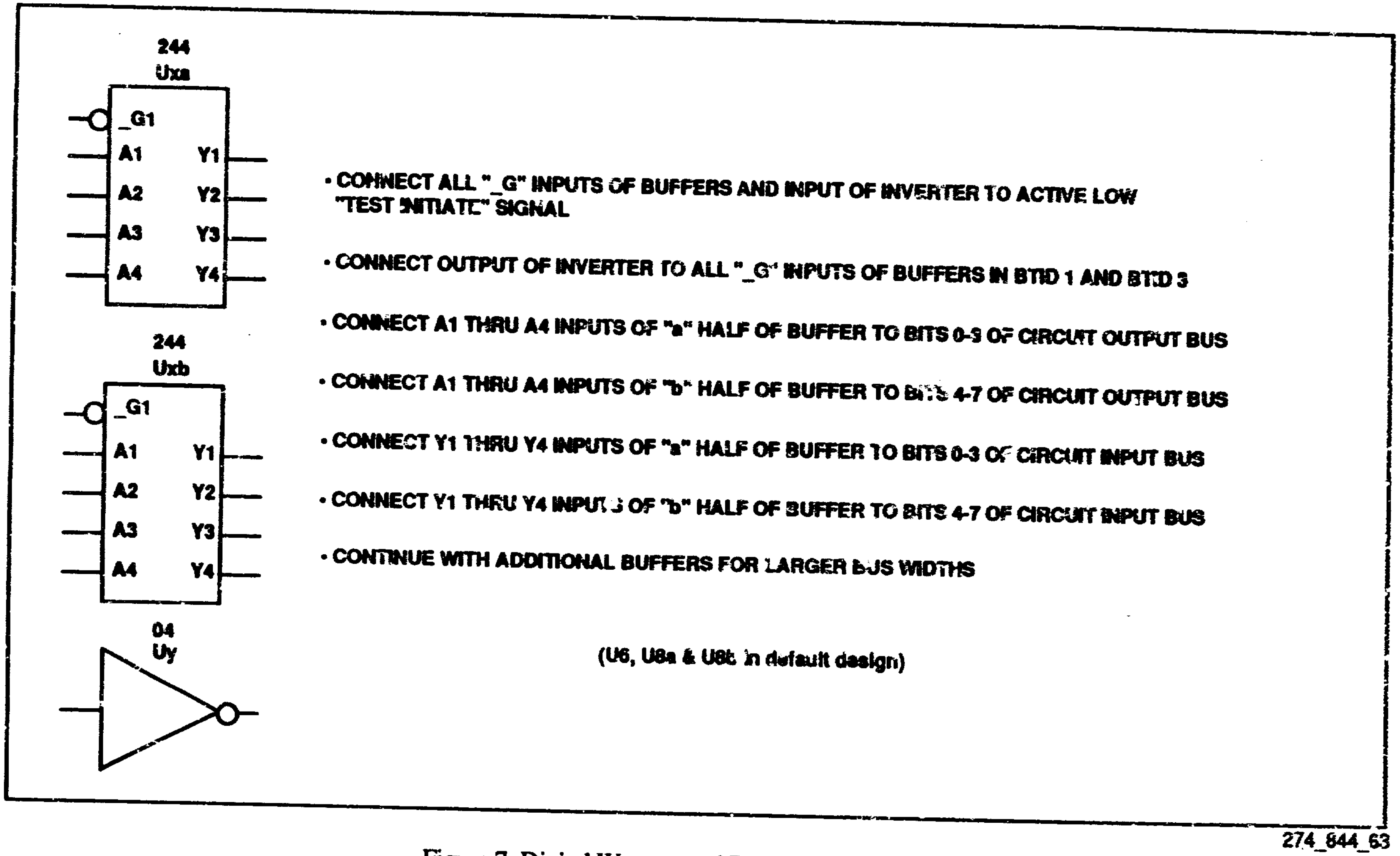


- CONNECT ALL "_G" INPUTS OF BUFFERS TO INVERTER OUTPUT OF BTID 2
- CONNECT A1 THRU A4 INPUTS OF "a" HALF OF BUFFER TO BITS 0-3 OF CIRCUIT OUTPUT BUS
- CONNECT A1 THRU A4 INPUTS OF "b" HALF OF BUFFER TO BITS 4-7 OF CIRCUIT OUTPUT BUS
- CONNECT Y1 THRU Y4 INPUTS OF "a" HALF OF BUFFER TO BITS 0-3 OF SYSTEM OUTPUT BUS
- CONNECT Y1 THRU Y4 INPUTS OF "b" HALF OF BUFFER TO BITS 4-7 OF SYSTEM OUTPUT BUS
- CONTINUE WITH ADDITIONAL BUFFERS FOR LARGER BUS WIDTHS

(U9a & U9b in default design)

274_844_32

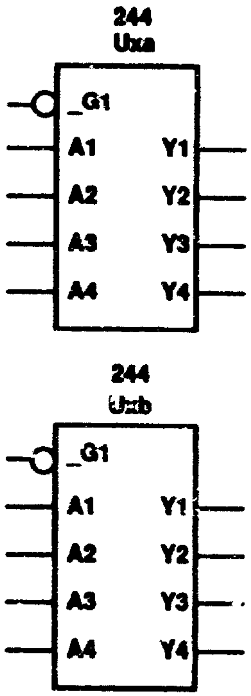
Figure 6 Digital Wraparound BTID 1 - Output Buffers



- CONNECT ALL "_G" INPUTS OF BUFFERS AND INPUT OF INVERTER TO ACTIVE LOW "TEST INITIATE" SIGNAL
- CONNECT OUTPUT OF INVERTER TO ALL "_G" INPUTS OF BUFFERS IN BTID 1 AND BTID 3
- CONNECT A1 THRU A4 INPUTS OF "a" HALF OF BUFFER TO BITS 0-3 OF CIRCUIT OUTPUT BUS
- CONNECT A1 THRU A4 INPUTS OF "b" HALF OF BUFFER TO BITS 4-7 OF CIRCUIT OUTPUT BUS
- CONNECT Y1 THRU Y4 INPUTS OF "a" HALF OF BUFFER TO BITS 0-3 OF CIRCUIT INPUT BUS
- CONNECT Y1 THRU Y4 INPUTS OF "b" HALF OF BUFFER TO BITS 4-7 OF CIRCUIT INPUT BUS
- CONTINUE WITH ADDITIONAL BUFFERS FOR LARGER BUS WIDTHS

(U6, U8a & U8b in default design)

Figure 7 Digital Wraparound BTID 2 - Digital Gates

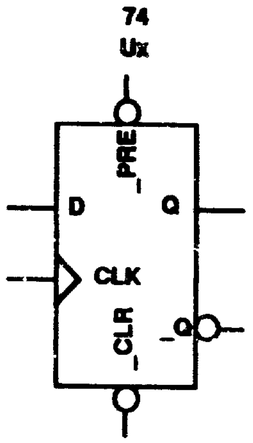


- CONNECT ALL "_G" INPUTS OF BUFFERS TO INVERTER CUTPUT OF BTID 2
- CONNECT A1 THRU A4 INPUTS OF "a" HALF OF BUFFER TO BITS 0-3 OF SYSTEM INPUT BUS
- CONNECT A1 THRU A4 INPUTS OF "b" HALF OF BUFFER TO BITS 4-7 OF SYSTEM INPUT BUS
- CONNECT Y1 THRU Y4 INPUTS OF "a" HALF OF BUFFER TO BITS 0-3 OF CIRCUIT INPUT CUS
- CONNECT Y1 THRU Y4 INPUTS OF "b" HALF OF BUFFER TO BITS 4-7 OF CIRCUIT INPUT BUS
- CONTINUE WITH ADDITIONAL BUFFERS FOR LARGER BUS WIDTHS

(U7a & U7b in default design)

274_844_84

Figure 8 Digital Wraparound BTID 3 - Input Buffers



- CONNECT "D" AND "_PRE" INPUTS OF FLIP FLOP TO Vcc
- CONNECT "CLK" INPUT OF FLIP FLOP TO RISING EDGE "TEST COMPLETE" SIGNAL
- CONNECT "_CLR" INPUT OF FLIP FLOP TO ACTIVE LOW "TEST INITIATE" SIGNAL
- CONNECT "Q" OUTPUT OF FLIP FLOP TO SYSTEM "PASS/FAIL STATUS" SIGNAL

(U10 In default design)

274_844_85

Figure 9 Digital Wraparound BTID 4 - Pass/fail Flip Flop

## BIT TECHNIQUE: DIGITAL WRAPAROUND

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | Number of CUT outputs to test | none |
| $v_2$. | #Test patterns for Digital Wraparound BIT | patterns |
| $v_3$. | Test pattern application rate for Digital Wrap BIT | patterns/sec |
| $v_4$. | CUT initialization time | sec |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 54ALS04B | INVERTER | 1 |
| 2 | 54ALS244A | BUFFER | 3 * ceil ( $v_1$ / b ) |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i = \text{ceil} ( u_i / upp_i )$$

**Explanation of symbols used:**

| | | |
|---|---|---|
| $n_i$ | = | Number of components (physical packages) required for i-th part |
| $u_i$ | = | Number of units (CAD symbols) required for i-th part |
| $upp_i$ | = | Number of units/package for i-th part (from Table 4.0) |
| ceil | = | Round up to nearest integer (e.g. 5.3 is rounded to 6) |
| b | = | Number of data bits for part (from Table 4.0) |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |

## BIT TECHNIQUE: DIGITAL WRAPAROUND

## CATEGORY: PENALTY EQUATIONS

**AREA (sq.in.)** = Sum ( $n_i * a_i$ ) + 15% for traces

**WEIGHT (gms)** = Sum ( $n_i * w_i$ ) + 10% for solder

**POWER (mW)** = Sum ( $n_i * p_i$ )

**TEST TIME (ns)** = ($v_4$ + tBUFFER) + ( $v_2 * (1/v_3)$ ) + ($v_4$ + tBUFFER)

**DELAY (ns)** = 2 * tBUFFER

**Explanation of Symbols Used.**

| | | |
|---|---|---|
| Sum | = | Sum over parts in BIT Technique Parts Table (i=1 to 2) |
| $n_i$ | = | Number of component packages for i-th part in Parts Table<br>$n_i$ is calculated according to Component Determination Equations |
| $a_i$ | = | Area in sq.in. of i-th part in Parts Table |
| $p_i$ | = | Power dissipation in mW (typical) for i-th part in Parts Table |
| $w_i$ | = | Weight in grams of i-th part in Parts Table |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |
| tBUFFER | = | Max delay for BUFFER from Table 4.0 |

## CATEGORY: BIBLIOGRAPHY

None required.

# PSEUDORANDOM PATTERN GENERATOR WITH MULTIPLE INPUT SIGNATURE REGISTER (PRPG/MISR)

## BIT TECHNIQUE DATA PACKAGE

**CATEGORY: SHORT DESCRIPTION / DIAGRAM**

This non-concurrent self test method can be implemented in hardware without requiring numerous test patterns or good machine responses to be stored internally. Testing begins upon activation of a test initiate signal, after which the test control logic initializes the Pseudorandom Pattern Generator (PRPG), which then generates and applies a set of pseudorandom test patterns to the Circuit Under Test (CUT) by multiplexing out the primary inputs and multiplexing in the PRPG outputs. A PRPG with n outputs will pseudorandomly cycle through all but one of the possible n - bit binary patterns, which is a total of $2^n - 1$ possible bit patterns. In order to determine if the CUT's response to these patterns is correct, the outputs of the CUT are connected in parallel to the Multiple Input Signature Register (MISR) with m inputs, which compresses the test result data into a single m - bit signature. If the test result signature is identical to the good machine signature, the test passes. This test method is advantageous because one achieves a substantial amount of testing with a relatively small amount of hardware.

Figures 1 and 2 show the Level I and II Block Diagrams for this BIT technique.

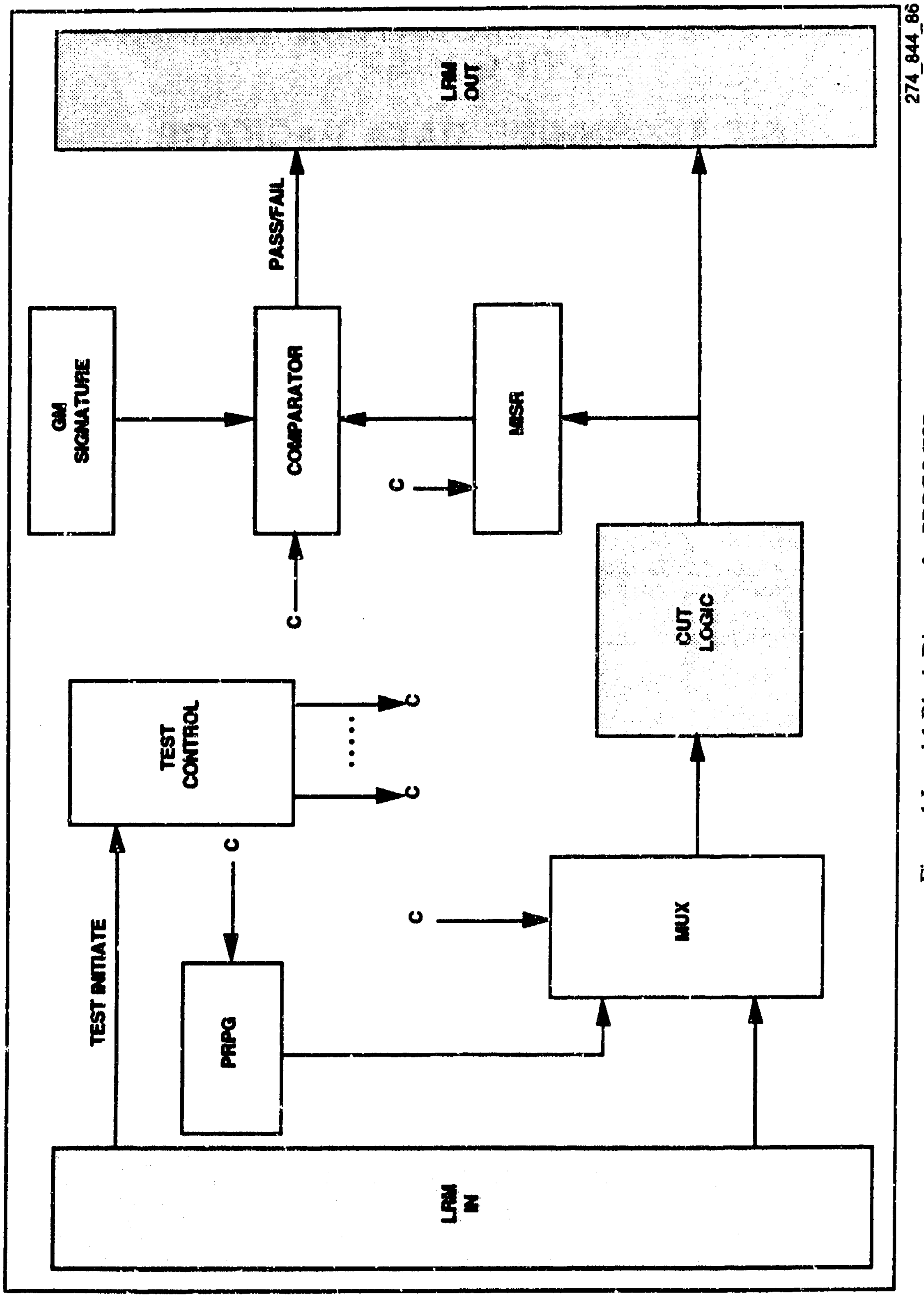


Figure 1 Level 1 Block Diagram for PRPG/MISR

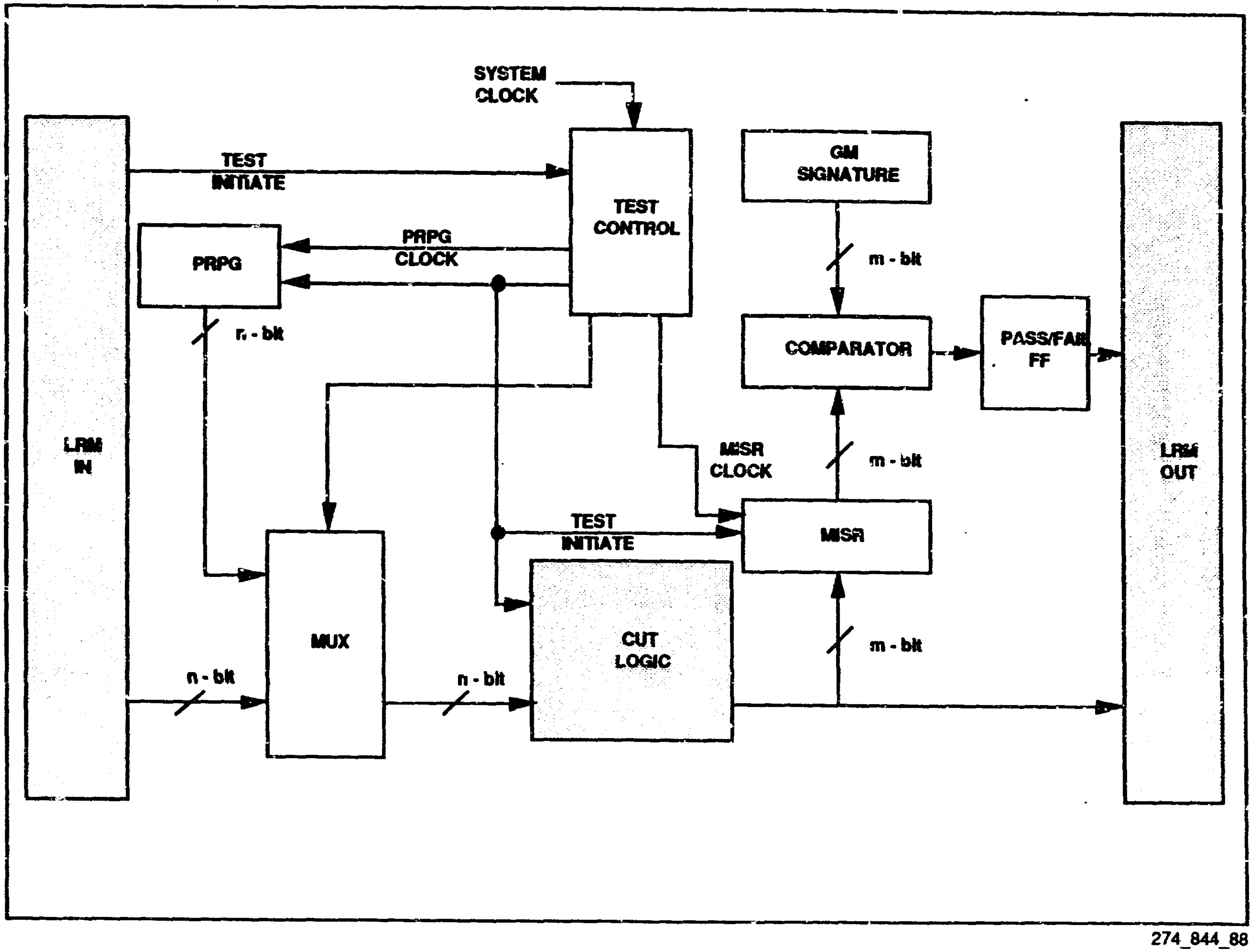


Figure 2 Level II Block Diagram For PRPG/MISR BIT Technique

**BIT TECHNIQUE: PRPG/MISR**

**CATEGORY: FLOW CHART / DESCRIPTION**

Figure 3 shows the Flow Chart for this BIT technique.

1. Receive BIT initiate and start test.
2. Initialize CUT, reset circuits, seed PRPG, select CUT inputs to be output of PRPG.
3. Clock Pseudorandom Pattern Generator and apply test pattern to CUT.
4. Test pattern ripples through CUT.
5. Clock multiple input signature register with falling edge of clock.
6. Apply next test pattern; continue until all patterns are applied.
7. Compare GM signature with CUT signature; set PASS/FAIL.
8. Select CUT inputs to be input from system.

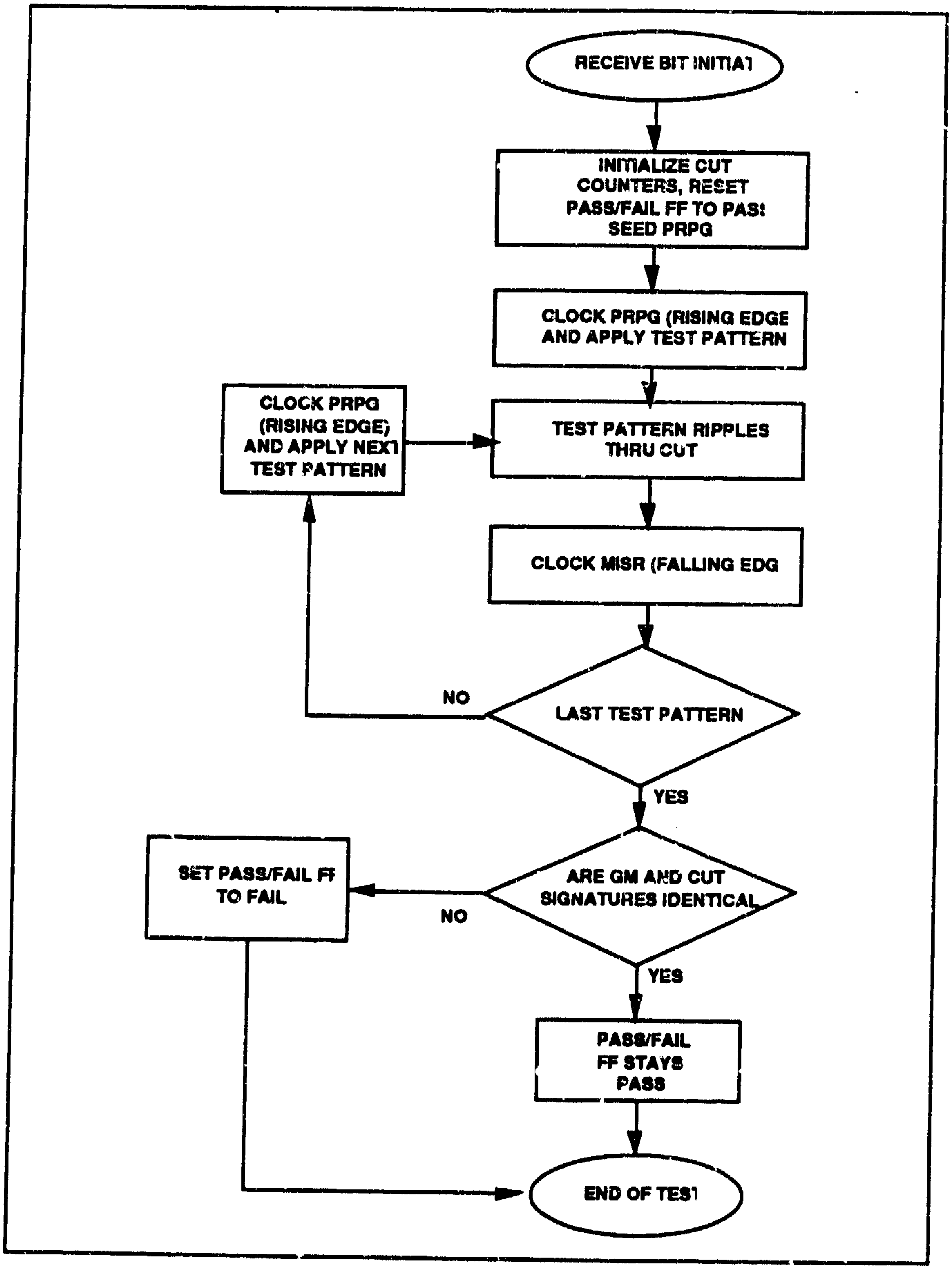

RECEIVE BIT INITIAT
INITIALIZE CUT COUNTERS, RESET PASS/FAIL FF TO PASS SEED PRPG
CLOCK PRPG (RISING EDGE AND APPLY TEST PATTERN
CLOCK PRPG (RISING EDGE) AND APPLY NEXT TEST PATTERN
TEST PATTERN RIPPLES THRU CUT
CLOCK MISR (FALLING EDG
NO
LAST TEST PATTERN
YES
SET PASS/FAIL FF TO FAIL
NO
ARE GM AND CUT SIGNATURES IDENTICAL
YES
PASS/FAIL FF STAYS PASS
END OF TEST
274_844_87

Figure 3 BIT Sequence Flow Chart For PRPG/MISR

## BIT TECHNIQUE: PRPG/MISR

### CATEGORY: ADVANTAGES

1. PRPG/MISR is a very cost effective BIT technique because large numbers of test patterns can be generated without the need for large hardware overhead.
2. There is no software overhead.
3. Multiple input signature registers allow multiple CUT outputs to be tested without storing huge good machine responses in ROM.

### CATEGORY: DISADVANTAGES

1. A PRPG cannot systematically generate specific pairs of test patterns in sequence to detect faults in certain sequential logic designs.
2. More pseudorandom patterns may be required to achieve a desired percentage of fault coverage than a set of test vectors that can be individually specified, such as in the On-Board ROM technique.

**BIT TECHNIQUE: PRPG/MISR**

**CATEGORY: ATTRIBUTES, Contd**

9. CONCEPTUAL COMPLEXITY

- Straightforward

10. HARDWARE/SOFTWARE/FIRMWARE

- Hardware

11. IS BITE SELF-TESTABLE?

- No

12. DESIGN COST

- Minimal with off the shelf chips available.

13. STAND-ALONE (SELF-CONTAINED) BIT?

- Yes

14. WEIGHT PENALTY

- Roughly proportional to area penalty.

## BIT TECHNIQUE: PRPG/MISR

## CATEGORY: DEFAULT DESIGN

This section describes the sequence of events for the PRPG/MISR BIT technique default design. Refer to Figure 4 for the default design schematic.

1. At the beginning of the test, the TEST_INITIATE signal goes low and triggers the following:

   a. Presets U1A test control flip-flop and enables TEST_CLOCK signal to activate counters, PRPG, and MISR.

   b. While the TEST_INITIATE signal holds low, the next rising edge of the TEST_CLOCK signal loads zeroes into the synchronous 4-bit binary counters U18 and U19.

   c. Disables normal inputs of CUT. Configures U16 and U17 MUX to select inputs from PRPG to CUT.

   d. Enables U20 8-bit identity comparator, disables U21A and U21B octal buffers and places the outputs of CUT into high impedance.

   e. For PRPG, presets U1B flip-flop and clears U3A, U3B, U4A, U4B, U5A, U5B, and U6A flip-flops.

   f. Resets all resetable logics in the CUT.

   g. For MISR, presets U6B flip-flop and clears U7A, U7B, U8A, U8B, U9A, U9B, and U10A flip-flops.

   h. Clears the Pass/Fail flip-flop U10B to PASS.

2. After the TEST_INITIATE signal goes back to high, the TEST_CLOCK signal starts incrementing counters U18 and U19. While the PRPG, CUT, and counters are driven directly by TEST_CLOCK, the MISR is driven by falling edge of the TEST_CLOCK to ensure enough delay time for the PRPG test pattern to ripple through the CUT before it reaches the MISR.

**BIT TECHNIQUE: PRPG/MISR**

**CATEGORY: DEFAULT DESIGN, Contd**

3. After 255 TEST_CLOCK cycles (2^8 - 1 = 255), U19/RCO outputs a high and indicates the end of the test:

   a. Clears U1A test control flip-flop and disconnects TEST_CLOCK signal to counters, PRPG, and MISR.

   b. Enables normal inputs of CUT. Configures U16 and U17 MUX to select inputs from normal inputs to CUT.

   c. Disables U20 8-bit identity comparator and enables U21A and U21B octal buffers for normal CUT outputs.

   d. Clocks in the result of the comparison of Good Machine Response and compressed signature. This result is registered by the Pass/Fail flip-flop U10B.

4. The circuit resumes normal operation and it will go into test mode when the TEST_INITIATE signal goes low again.

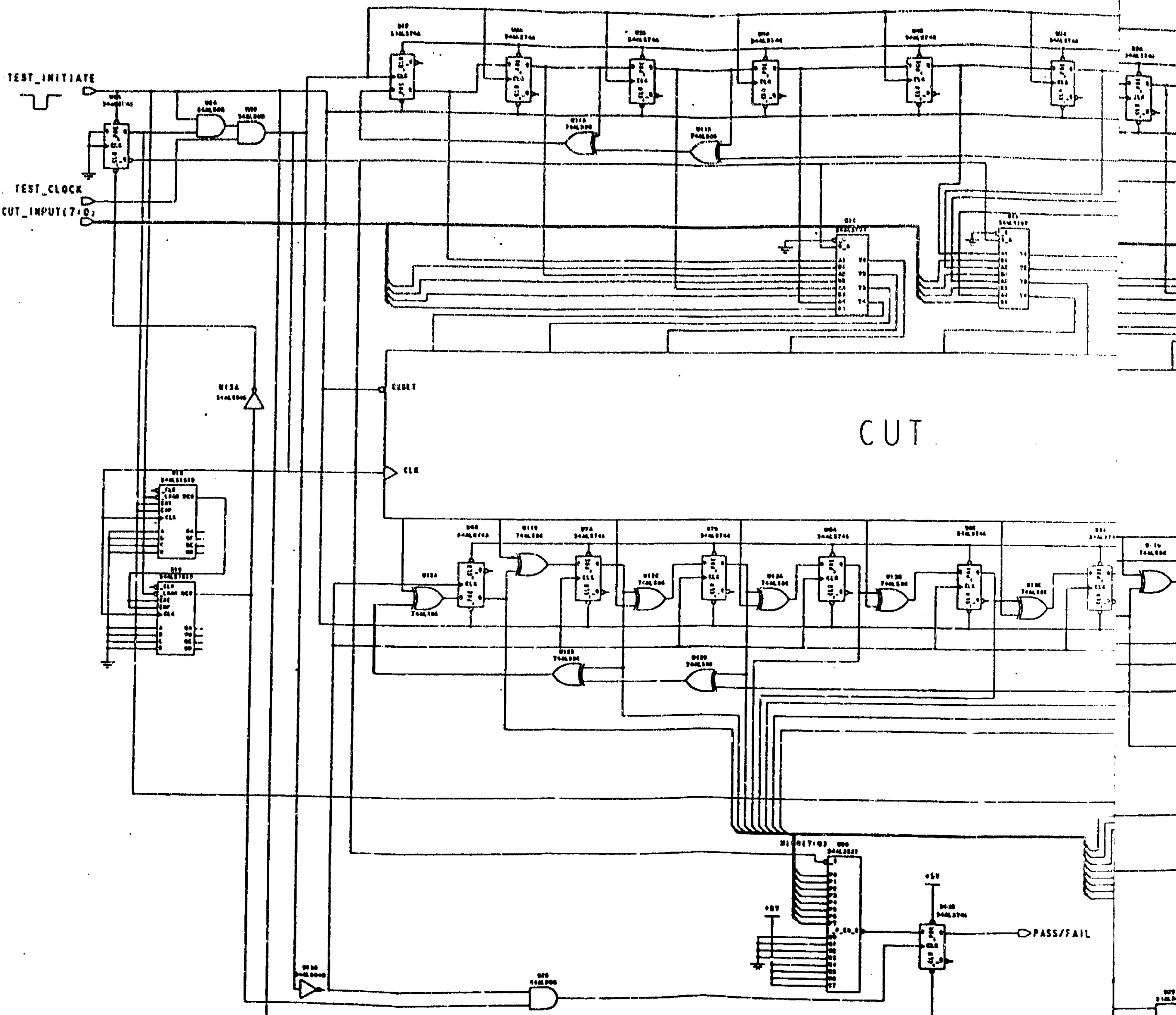


Figure 4 PRPG/MISR – De

1

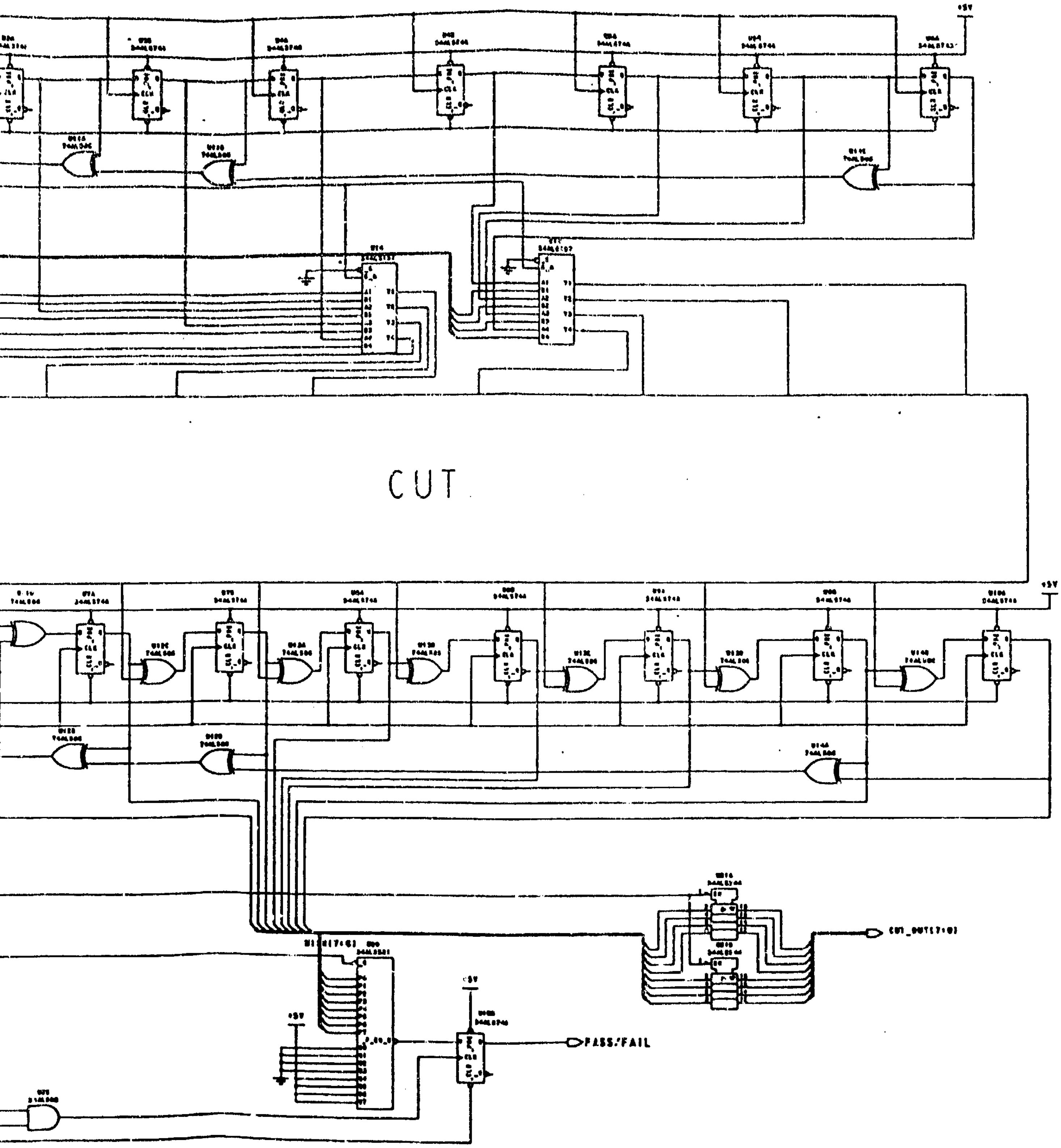


Figure 4 PRPG/MISR – Default Design

**BIT TECHNIQUE: PRPG/MISR**

**CATEGORY: BIT TECHNIGUE INSERTION DIAGRAM**

Figure 5 shows the BIT Technique Insertion Diagram for this BIT technique.

BIT ELEMENT 1:

1. Connect the control gates and Flip Flop to the control counters.
2. Connect the LRM inputs to the MUX inputs.
3. Connect the MUX outputs to the CUT inputs

BIT ELEMENT 2:

1. Connect the seed counter output to the inputs of PRPG.
2. Connect the output of PRPG to the CUT inputs.

BIT ELEMENT 3 :

1. Connect the CUT outputs to the input of MISR.

BIT ELEMENT 4 :

1. Connect the the output of MISR to the comparator.
2. Connect the output of the comparator to the input of the D input of the Pass/Fail Flip Flop.
3. Connect the Reset signal to the Pass/Fail Flip Flop CLR signal line.

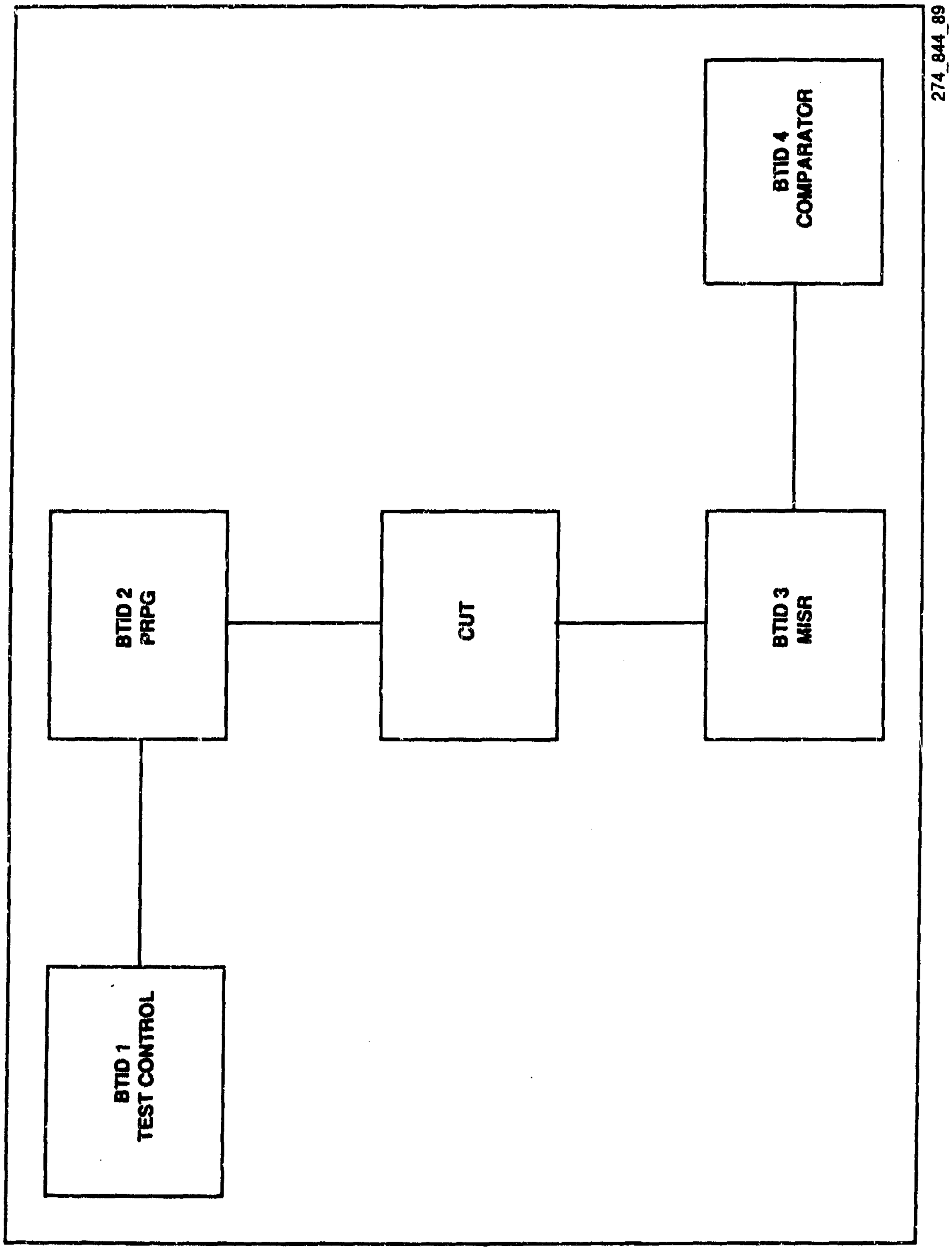


Figure 5 BIT Technique Insertion Diagram For PRPG/MISR BIT Technique

## BIT TECHNIQUE: PRPG / MISR

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | Number of CUT inputs | none |
| $v_2$. | Number of CUT outputs to test | none |
| $v_3$. | Maximum propagation delay through CUT | ns |
| $v_4$. | CUT initialization time | sec |
| v5. | Internal Design OK? (Suitability Attribute) | 0 if YES, 1 if NO |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 54ALS04B | INVERTER | 2 |
| 2 | 54ALS08 | AND GATES | 3 |
| 3 | 54ALS74A | FLIP-FLOP | $1 + v_1 + (v_5 * v_2) + 1$ |
| 4 | 54ALS86 | EXCLUSIVE-OR | $3 + (v_2 + 3)$ |
| 5 | 54ALS157 | SELECTOR/MUX | $\text{ceil}\ ( v_1 / b )$ |
| 6 | 54ALS161B | COUNTER | $\text{ceil}\ ( v_1 / b )$ |
| 7 | 54ALS244A | BUFFER | $\text{ceil}\ ( v_2 / b )$ |
| 8 | 54ALS521 | COMPARATOR | $\text{ceil}\ ( v_2 / b )$ |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i = \text{ceil}\ ( u_i / upp_i )$$

**Explanation of symbols used:**

| | | |
|---|---|---|
| $n_i$ | = | Number of components (physical packages) required for i-th part |
| $u_i$ | = | Number of units (CAD symbols) required for i-th part |
| $upp_i$ | = | Number of units/package for i-th part (from Table 4.0) |
| ceil | = | Round up to nearest integer (e.g. 5.3 is rounded to 6) |
| b | = | Number of data bits for part (from Table 4.0) |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |

## BIT TECHNIQUE: PRPG / MISR

## CATEGORY: PENALTY EQUATIONS

**AREA (sq.in.)** = Sum ( $n_i * a_i$ ) + 15% for traces

**WEIGHT (gms)** = Sum ( $n_i * w_i$ ) + 10% for solder

**POWER (mW)** = Sum ( $n_i * p_i$ )

**TEST TIME (ns)** = Max ($v_4 * 10^9$, 50) + ( p * 2M * (tMISR + $v_3$) ) + tCOMP + tFF

**DELAY (ns)** = tMUX + tXOR + $v_5$*tFF + tBUFFER

**Explanation of Symbols Used.**

Sum = Sum over parts in BIT Technique Parts Table (i=1 to 8)

$n_i$ = Number of component packages for i-th part in Parts Table
$n_i$ is calculated according to Component Determination Equations

$a_i$ = Area in sq.in. of i-th part in Parts Table

$p_i$ = Power dissipation in mW (typical) for i-th part in Parts Table

$w_i$ = Weight in grams of i-th part in Parts Table

$v_i$ = User-supplied value for i-th variable (see Variable Definitions)

M = Design margin = 1.2

p = Number of test patterns = $2^{v1}$ - 1

tMISR = MISR time = tFF + tXOR

tMUX = Max delay for SELECTOR/MUX from Table 4.0

tCOMP = Max delay for COMPARATOR from Table 4.0

tFF = Max delay for FLIP-FLOP from Table 4.0

tXOR = Max delay for EXCLUSIVE-OR from Table 4.0

tBUFFER = Max delay for BUFFER from Table 4.0

**BIT TECHNIQUE: PRPG/MISR**

**CATEGORY: BIBLIOGRAPHY**

BILBO - Built-In Logic Block Observation Techniques
79 - Koenemann, Mucha, Zwieoff - 1979 IEEE Test Conference

81 - Segers - 1981 IEEE Test Conference -
A Self-Test Method for Digital Circuits

STUMPS - Self Testing of Multi Logic Modules
82 - Bardell, McAnney - 1982 IEEE Test Conference

83 - Komonytsky - Electronics 1983 -
Synthesis of Techniques Creates Complete System Self-Test

84 - Butt, El-ziq - 1984 International Test Conference - Impact of Mixed-Mode Self-Test On Life Cycle Cost of VLSI Based Designs

84 - LeBlanc - 1984 IEEE Design & Test of Computers
LOCST: A Built-In Self-Test Technique

85 - Bhavsar - 1985 International Test Conference -
"Concatenable Polydividers': BIT-Sliced LFSR Chips For Board Self-Test

86 - Sabe, Johannsen, Yau - 1986 Custom Integrated Circuits Conference - Genesil Silicon Compilation and Design for Testability

# COMPARATOR

# BIT TECHNIQUE DATA PACKAGE

## CATEGORY: SHORT DESCRIPTION

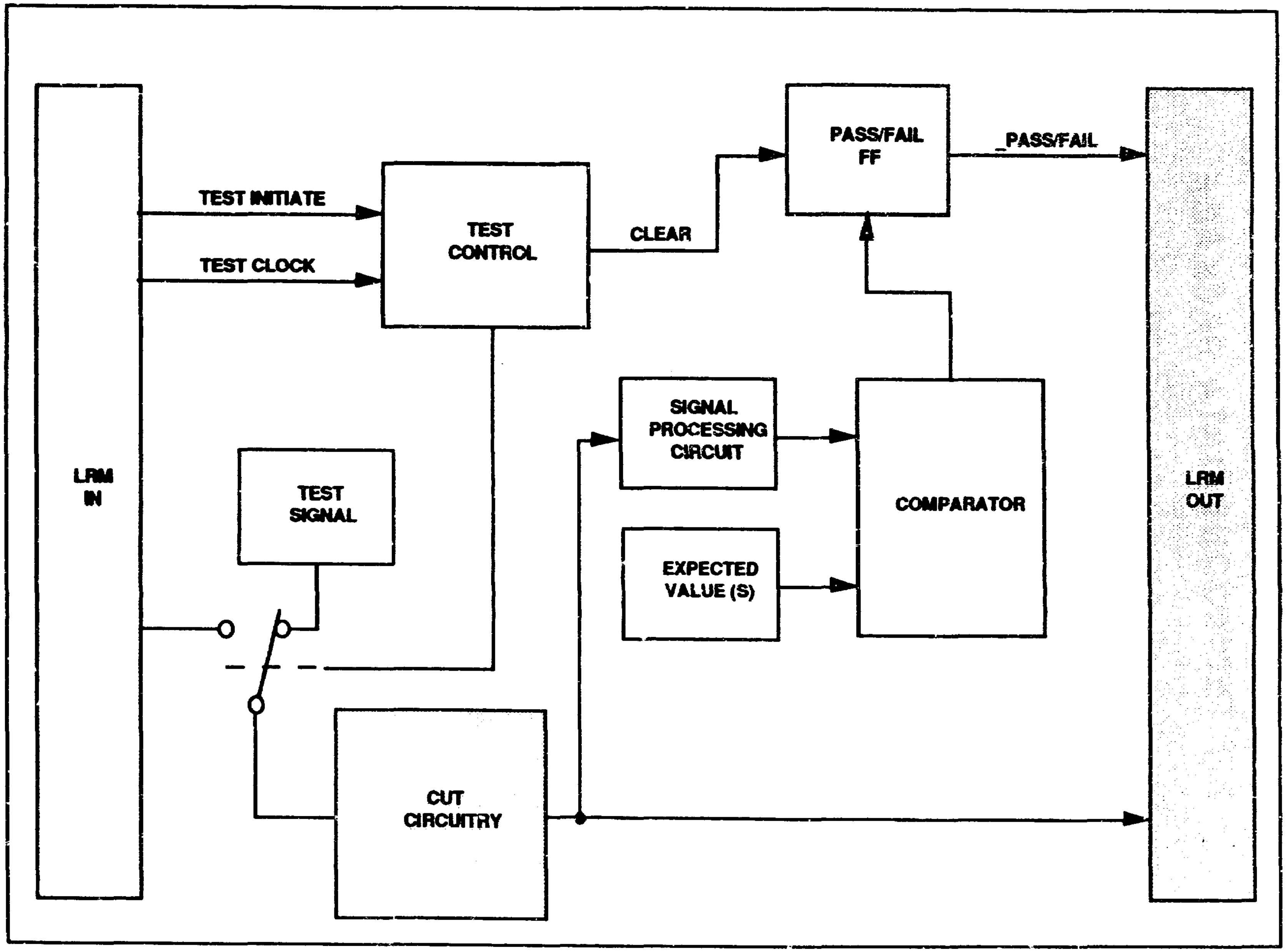
Comparators can readily be incorporated into hardware designs to achieve Built-In-Test (BIT) capability for a large variety of functions with minimum expense. With this approach, Circuit Under Test (CUT) stimuli, which are generated by the Line Replaceable module (LRM), are applied to a CUT and the outputs of the CUT are applied to a comparator along with a reference signal. The outputs of the CUT are compared to the reference signal and the comparator outputs a PASS/FAIL signal. For some applications, it will be necessary to process the CUT outputs with the addition of signal processing circuitry before feeding the result to the comparator.

The Comparator BIT technique allows for either including a signal source as part of the BIT hardware or for receiving the test signal from outside the LRM. If multiple channels are present on the CUT, analog switches can be added to distribute test signals to various channel inputs and analog multiplexers to distribute the CUT outputs to the comparator for analysis.

Due to the wide variety of processing circuits available (e.g., frequency to voltage converters, RMS to dc, peak detectors, etc.), together with benefits gained from signal multiplexing, the Comparator BIT technique lends itself to a wide variety of applications.

Due to the wide variety of processing circuits available, (e.g. frequency to voltage converters, sample and hold circuits), together with benefits gained from signal multiplexing, the COMPARATOR BIT TECHNIQUE lends itself to a wide variety of applications.

Figures 1 and 2 show the Level I and II Block Diagrams for this BIT technique.



274_844_90

Figure 1 Level I Block Diagram Utilizing Comparator As A Bit Technique

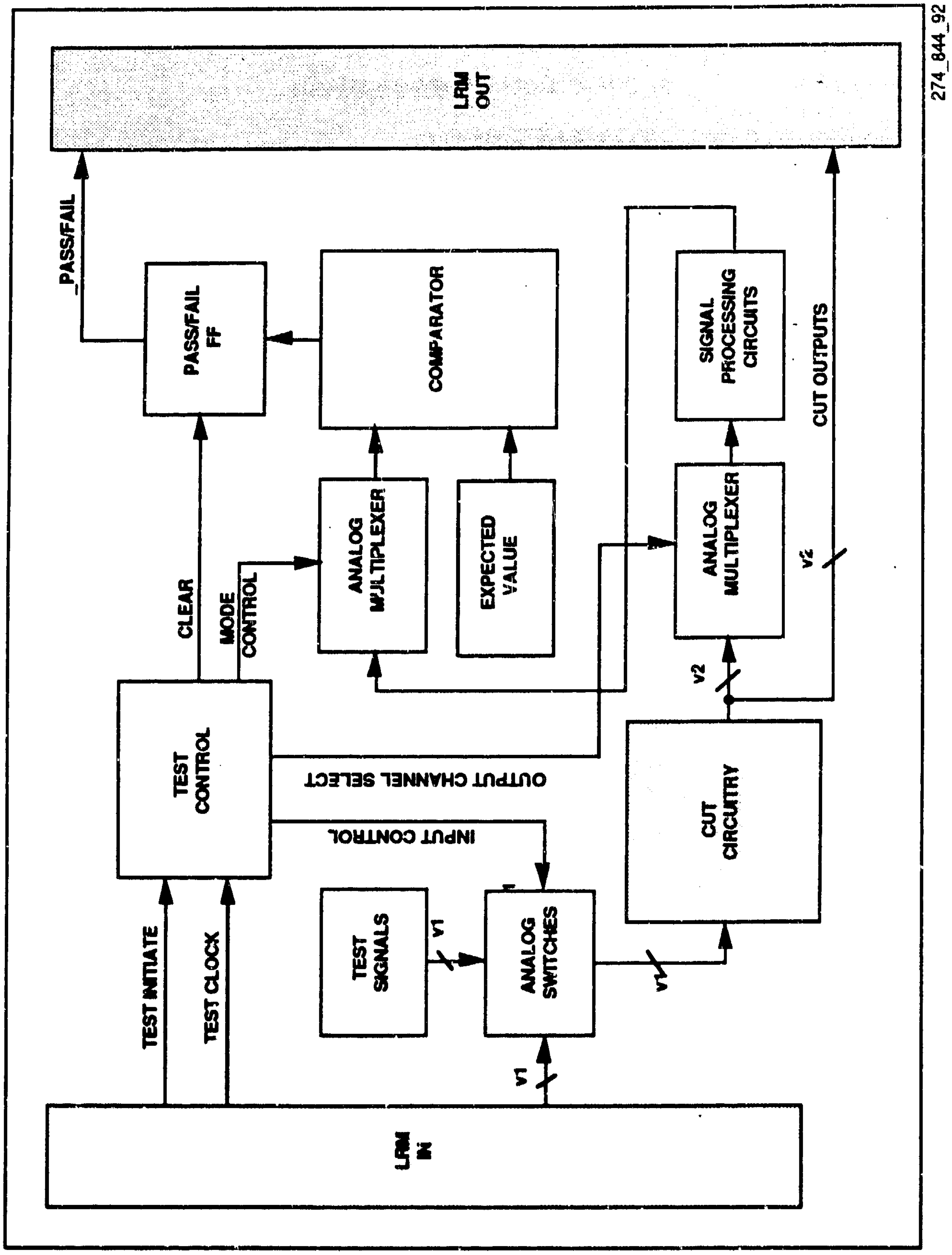


Figure 2 Level II Block Diagram Utilizing Comparator As A BIT Technique

## BIT TECHNIQUE: COMPARATOR

## CATEGORY: FLOW CHART / DESCRIPTION

Figure 3 shows the Flow Chart for this BIT technique.

1. Initiate test by sending a negative going pulse to the TEST INITIATE line.

2. Set the counter to zero. Set the Pass/Fail FF to PASS.

3. Apply the test signals to the CUT.

4. Select the CUT output to be processed or fed directly to the comparator.

5. If required, process the output by converting the parameter of interest (AC, RMS, frequency, peak value, etc.) to a DC value. Route the signal to the comparator.

6. If the signal falls outside the limits of the comparator, set the PASS/FAIL FF to FAIL.

7. Determine if output is the last one to be tested. If it is, end the test.

8. If the last CUT output has not been tested, increment the counter and repeat Steps 4 through 7.

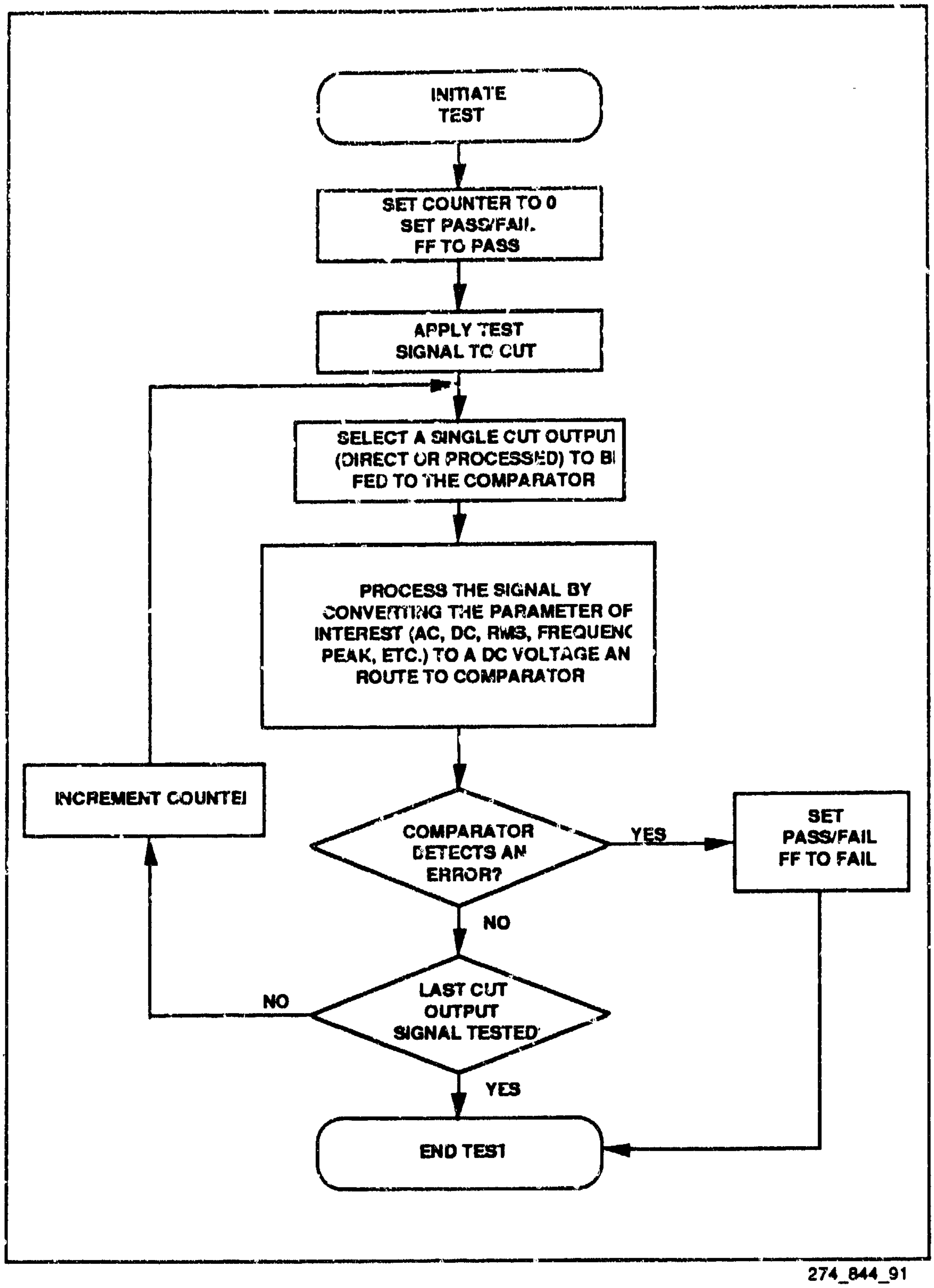

INITIATE TEST
SET COUNTER TO 0
SET PASS/FAIL
FF TO PASS
APPLY TEST SIGNAL TO CUT
SELECT A SINGLE CUT OUTPU
(DIRECT OR PROCESSED) TO B
FED TO THE COMPARATOR
PROCESS THE SIGNAL BY CONVERTING THE PARAMETER OF INTEREST (AC, DC, RMS, FREQUENC PEAK, ETC.) TO A DC VOLTAGE AN ROUTE TO COMPARATOR
INCREMENT COUNTE
COMPARATOR DETECTS AN ERROR?
YES
SET PASS/FAIL FF TO FAIL
NO
NO
LAST CUT OUTPUT SIGNAL TESTED
YES
END TEST
274_844_91

Figure 3 BIT Sequence Flow Chart For Utilizing Comparator Testing Techniques For N Channels Or Signals

## BIT TECHNIQUE: COMPARATOR

## CATEGORY: ADVANTAGES

- The Comparator BIT can be used to verify the amplitude and frequency of many kinds of signals (DC, AC, RF, AM, and FM modulated signals, digital pulses, and others) if appropriate signal processing is provided.
- An unlimited number of test point signals can be monitored by one circuit with the use of multiplexers.
- Circuit components are common and can be purchased off the shelf from a wide variety of vendors.
- The high input impedance of the window comparator and scaling circuits place very little load on circuit outputs.

## CATEGORY: DISADVANTAGES

- The reference signal must be accurately maintained. Any deviation from the desired reference could cause erroneous BIT results.
- Frequently, additional power supplies are required as either op amp power supplies or as reference voltage supplies.
- The analog switches used to switch the CUT input lines between the test signals and the actual LRM signals add 75Ω of on resistance to the CUT inputs which could slow down these inputs.
- CUT outputs that are to be BIT tested must be scaled to fit within the window of the comparator, i.e. they must have the same tolerance band as that of the comparator.
- BIT circuitry requires a clock signal to control channel selection and to allow proper settling of analog signals before measurement is made.
- For static (dc) test signals, resistor values must be selected. For more complicated test signals, circuitry must be added to generate those signals.

**BIT TECHNIQUE: COMPARATOR**

**CATEGORY: ATTRIBUTES**

1. CONCURRENCY

- Nonconcurrent

2. TECHNOLOGY

- Typically applicable to circuits using current ANALOG technology.
- Existing off-the-shelf analog and digital integrated circuits are used for the BIT circuit.

3. CUT MICROPROCESSOR REQUIRED?

- No

4. CUT INTERNAL DESIGN REQUIRED?

- BIT circuitry is external to the CUT.

5. AREA PENALTY

- The use of signal multiplexing with the comparator technique reduces the real estate penalty incurred in contrast to the use of concurrent techniques such as redundancy. This is particularly true if large numbers of circuits are to be BIT tested.
- If signal processing circuits are required, the real estate penalty increases proportionately.
- Required real estate for implementation of the technique increases as the number of CUT signals tested increases. The amount of real estate required can be minimized with the use of multiplexers.
- The need for resistors and capacitors adds to the real estate penalty.
- The addition of gating circuitry may be necessary for enabling and latching.

**BIT TECHNIQUE: COMPARATOR**

**CATEGORY: ATTRIBUTES, Contd**

6. WEIGHT PENALTY

- The use of signal multiplexing reduces the weight penalty compared with concurrent techniques. This is particularly true if large numbers of circuits are to be BIT tested.
- If signal processing circuits are required, the weight penalty increases.

7. POWER PENALTY

- Signal multiplexing reduces the power penalty over concurrent techniques.
- If signal processing circuits are required, the power penalty increases proportionately.

8. TIMING PENALTY

- Throughput delay
- During normal operation, the Analog Switches placed in series with the CUT input lines add an on-resistance of up to 75Ω, which could affect CUT input circuit speed.
- Test Time
- During BIT, normal operation of the CUT is suspended while the CUT is being tested.

9. RELIABILITY IMPACT

- The failure rate goes up proportionately with the number of components added for the BIT and signal processing circuitry.
- The input switches are the only in-line components that could cause a critical system failure.

**BIT TECHNIQUE: COMPARATOR**

**CATEGORY: ATTRIBUTES, Contd**

10. CONCEPTUAL COMPLEXITY

- Straightforward.

11. HARDWARE/SOFTWARE/FIRMWARE

- BIT circuitry uses analog switches and multiplexers, comparators, and digital logic to implement the BIT technique. If signal processing circuits are required, many other types of components may be necessary.
- No software required on the board level.
- Simple system software required to reset, initiate, and monitor the BIT circuit.

12. DESIGN COST

- The use of the Comparator BIT technique reduces design costs compared to concurrent techniques(use of redundant comparator circuits) if large numbers of circuits are to be BIT tested.
- If signal processing circuits are required, the design costs increase proportionately.

13. MEMORY REQUIREMENTS

- None

14. BIT CIRCUITRY SELF-TESTABLE?

- Yes. Unused channels of the output multiplexer, along with additional divider network resistors, can be utilized to provide test inputs to the comparator.

15. STAND-ALONE (SELF-CONTAINED) BIT?

- Yes. BIT circuit is co-located with CUT on a common PCB. TEST INITIATE and PASS/FAIL signals are the only lines that interface the BIT to the LRM .

16. NOTES

- Complexity, cost, area, weight, and power are dependent on the number of signal processing circuits used. Default design does not use any.
-

## BIT TECHNIQUE: COMPARATOR

## CATEGORY: DEFAULT DESIGN

This section describes the Default Design for the Comparator BIT technique. Refer to Figure 4 for the Default Design schematic.

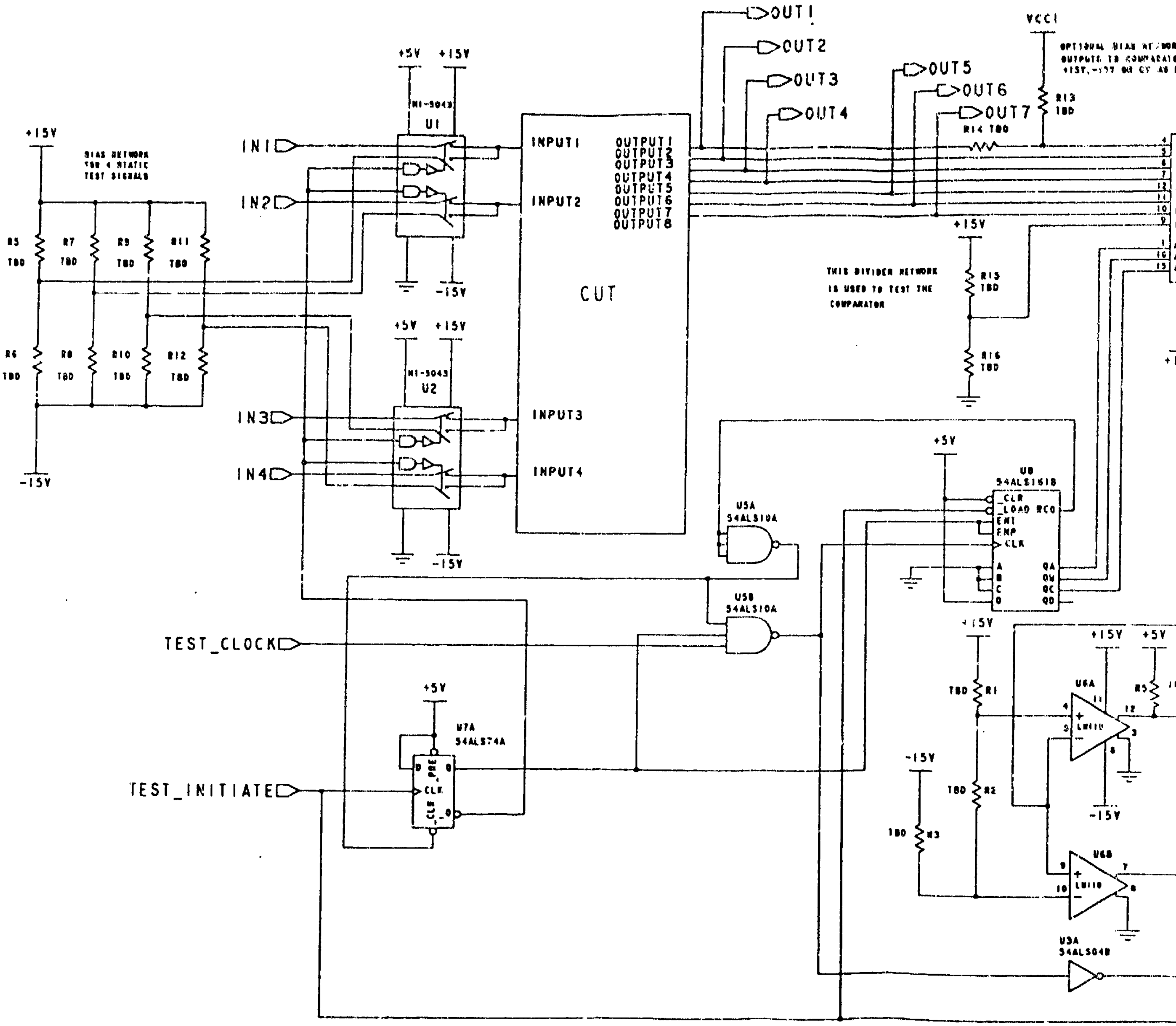
Functionally, the design uses analog switches (U1 and U2) to select between the normal CUT inputs and special test inputs. An analog multiplexer (U4) is used to select the particular CUT output to be measured. A window detector (U6) checks the signal to determine if it is within its tolerance limits. A counter (U8) generates the analog multiplexer channel select signals, and a PASS/FAIL latch (U5 and U7B) indicate the BIT test results.

In the Default Design, implemented resistor divider networks provide 4 static test inputs to the CUT. Two Harris HI-5043 (U1 and U2) analog switches provide the control to switch the CUT inputs between the LRM and the test inputs. The CUT outputs are scaled by resistor networks similar to R13 and R14 to fit within the limits of the window detector. These scaled outputs are fed to a Harris HI-508 (U4) analog multiplexer whose output feeds the LM119 (U6) window detector. The output is checked to limits determined by resistors R1 through R3. The wired AND outputs of the window detector are fed to a latchable FF to generate the BIT PASS/FAIL output. A 54ALS161B (U8) counter provides the channel select signals to the analog multiplexer. FF U7A is used to control the input select of the analog switches, U1 and U2. NAND gates U5A and U5B are used to stop the test once all CUT outputs have been tested. Resistor divider network R15 and R16 provide a test signal to the window detector as a self check of the BIT circuit.

Circuit Enhancements: The test signal resistor dividers (R5-R12) can be replaced by general purpose signal generators such as sine wave generators, square wave generators, etc., to create any type of test signal desired. Also, control loops can be tested by taking their output and electronically simulating the external hardware to close the loop. Additional analog switches can be added to switch more inputs. Also, more analog multiplexers can be added to look at more CUT outputs. Signal processors such as RMS/DC, AC/DC, frequency/DC, peak/DC, etc., wired in parallel from the analog multiplexer output would be used to convert any parameter of interest from the signal.

Another analog multiplexer would route the parameter of interest to the system A-to-D converter whose output would then be stored in memory. The window detector function would be done in software where the limits would be tailored to fit the signal being measured. PASS/FAIL information could also be extracted in software.

System Interface: Three signals, TEST_CLOCK, TEST_INITIATE, and _PASS/FAIL, are used to interface the BIT circuit to the system. The TEST_CLOCK signal is used to cycle the analog multiplexer through all of the CUT output channels. Its period is set large enough to accommodate the settling time of the BIT circuit. TEST_INITIATE is a positive pulse used to start the BIT test. _PASS/FAIL is used to inform the system of the results of the BIT test.


OUT1
OUT2
OUT3
OUT4
OUT5
OUT6
OUT7
VCC1
OPTIONAL BIAS NETWORK
OUTPUTS TO COMPARATOR
+15V,-15V OR CV AS
R13
TBD
R14 TBD
+5V
+15V
HI-5043
U1
+15V
BIAS NETWORK
FOR 4 STATIC
TEST SIGNALS
IN1
IN2
INPUT1
INPUT2
OUTPUT1
OUTPUT2
OUTPUT3
OUTPUT4
OUTPUT5
OUTPUT6
OUTPUT7
OUTPUT8
CUT
R5
TBD
R7
TBD
R9
TBD
R11
TBD
-15V
R6
TBD
R8
TBD
R10
TBD
R12
TBD
-15V
+5V
+15V
HI-5043
U2
IN3
IN4
INPUT3
INPUT4
-15V
+15V
THIS DIVIDER NETWORK
IS USED TO TEST THE
COMPARATOR
R15
TBD
R16
TBD
+5V
U8
54ALS161B
_CLR
_LOAD RCO
ENT
ENP
CLK
A
B
C
D
QA
QB
QC
QD
U5A
54ALS10A
U5B
54ALS10A
TEST_CLOCK
+5V
U7A
54ALS74A
D
PRE
Q
CLK
CLR
Q
TEST_INITIATE
+15V
TBD
R1
TBD
R2
-15V
TBD
R3
+15V
+5V
U6A
LM119
R5
-15V
U6B
LM119
U3A
54ALS04B

Figu

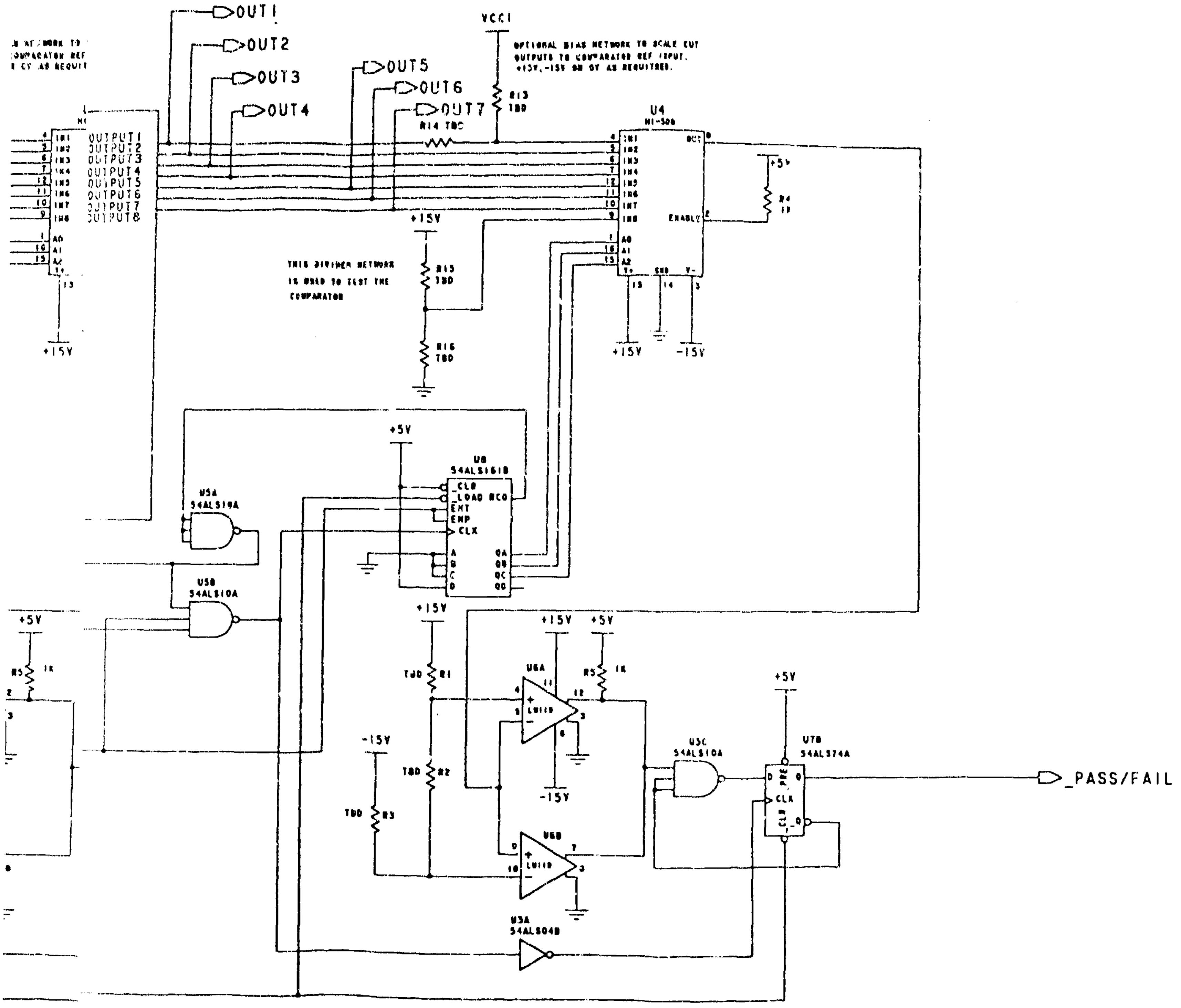


Figure 4

Figure 4 Comparator -- Default Design

## BIT TECHNIQUE: COMPARATOR

## CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM

Figure 5 shows the BIT Technique Insertion Diagram for this BIT technique.

BIT ELEMENT 1

1. Connect up to 4 LRM outputs to the analog switches (U1 and 2 pins 10 and 16).

2. Select resistors R5-R12 or use default valves to provide desired test signal for the CUT.

BIT ELEMENT 2

1. Connect the LRM TEST INITIATE output to the BIT circuit TEST INITIATE input (U7A clock).

2. Connect the LRM TEST CLOCK output to the BIT circuit TEST CLOCK input (U5B).

BIT ELEMENT 3

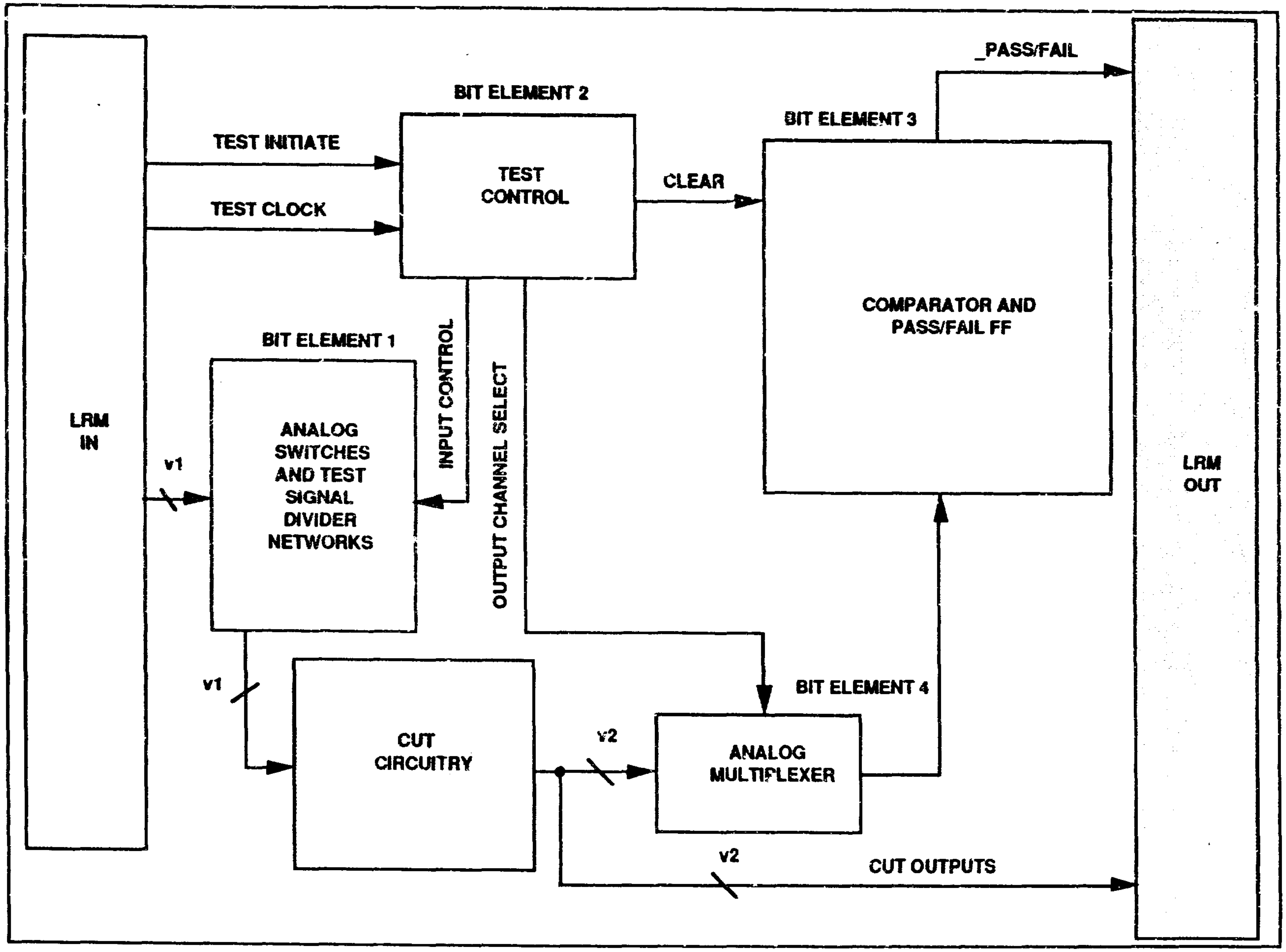
1. Select R1-R3 or use default values to provide the comparator nominal voltage and tolerance range for the CUT outputs to be tested.

2. Route the __PASS/FAIL BIT circuit output to the LRM microprocessor interrupt or suitable BIT error indicator.

BIT ELEMENT 4

1. Select R13 and R14 to scale CUT output 1 to match the comparator nominal voltage and tolerance range and connect to the analog multiplexer. As required, select other bias resistors (not shown) for remaining CUT outputs and connect to the analog multiplexers.

2. Select R15 and R16 or use default values to provide a test voltage to self test the BIT circuit.



274_844_93

Figure5 BTID Block Diagram Utilizing Comparator as a BIT Technique

## BIT TECHNIQUE: COMPARATOR

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | Number of CUT inputs | none |
| $v_2$. | Number of CUT outputs to test | none |
| $v_3$. | Test time per monitored voltage for Comparator BIT | microseconds |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 54ALS04B | INVERTER | 1 |
| 2 | 54ALS10A | NAND GATES | 3 |
| 3 | 54ALS74A | FLIP-FLOP | 2 |
| 4 | 54ALS161B | COUNTER | ceil ( $\log_2(v_2+1)$ / b ) |
| 5 | LM119 | COMPARATOR | 2 |
| 6 | HI-508 | MUX | ceil ( $v_2+1$ / b ) |
| 7 | HI-5043 | SWITCH | ceil ( $v_1$ / b ) |
| 8 | RNC55H | RESISTOR | $2v_1 + 4 + (2v_2 + 3)$ |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i \quad = \quad \text{ceil} ( u_i / upp_i )$$

**Explanation of symbols used:**

$n_i$ = Number of components (physical packages) required for i-th part
$u_i$ = Number of units (CAD symbols) required for i-th part
$upp_i$ = Number of units/package for i-th part (from Table 4.0)
ceil = Round up to nearest integer (e.g. 5.3 is rounded to 6)
b = Number of data bits for part (from Table 4.0)
$v_i$ = User-supplied value for i-th variable (see Variable Definitions)

## BIT TECHNIQUE: COMPARATOR

## CATEGORY: PENALTY EQUATIONS

**AREA (sq.in.)** = Sum ( $n_i$ * $a_i$ ) + 15% for uraces

**WEIGHT (gms)** = Sum ( $n_i$ * $w_i$ ) + 10% for solder

**POWER (mW)** = Sum ( $n_i$ * $p_i$ )

**TEST TIME (ns)** =

**DELAY (ns)** =

**Explanation of Symbols Used.**

Sum = Sum over parts in BIT Technique Parts Table (i=1 to 8)
$n_i$ = Number of component packages for i-th part in Parts Table
$n_i$ is calculated according to Component Determination Equations
$a_i$ = Area in sq.in. of i-th part in Parts Table
$p_i$ = Power dissipation in mW (typical) for i-th part in Parts Table
$w_i$ = Weight in grams of i-th part in Parts Table
$v_i$ = User-supplied value for i-th variable (see Variable Definitions)

## CATEGORY: BIBLIOGRAPHY

None required.

# VOLTAGE SUMMING

# BIT TECHNIQUE DATA PACKAGE

## CATEGORY: SHORT DESCRIPTION

Voltage summing is a concurrent analog Built-in-Test (BIT) technique whereby multiple voltage levels are added together using operational amplifiers. The resulting sum is then fed into a window comparator circuit which compares the sum against a reference signal(s). The output of this window comparator circuit generates a pass/fail signal. This technique is particulary useful for monitoring a set of power supply voltages.

This BIT technique is often used along with the comparator technique to test circuits with multiple output channels. Voltage summing BIT can also be used in conjunction with redundancy BIT techniques.

Figures 1 and 2 show the Level I and II Block Diagrams for this BIT technique.

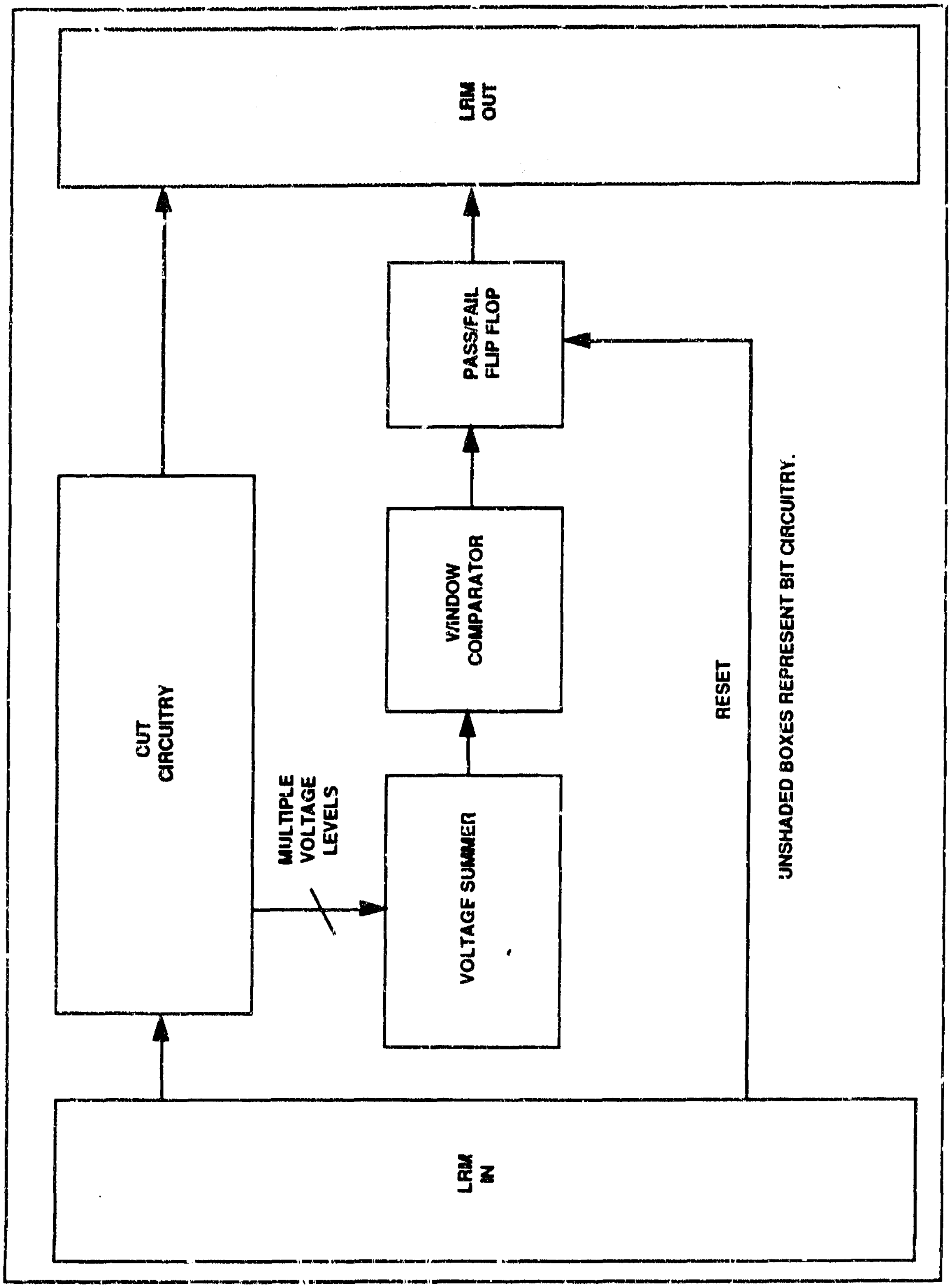


Figure 1 Level I Block Diagram for Voltage Summing Technique

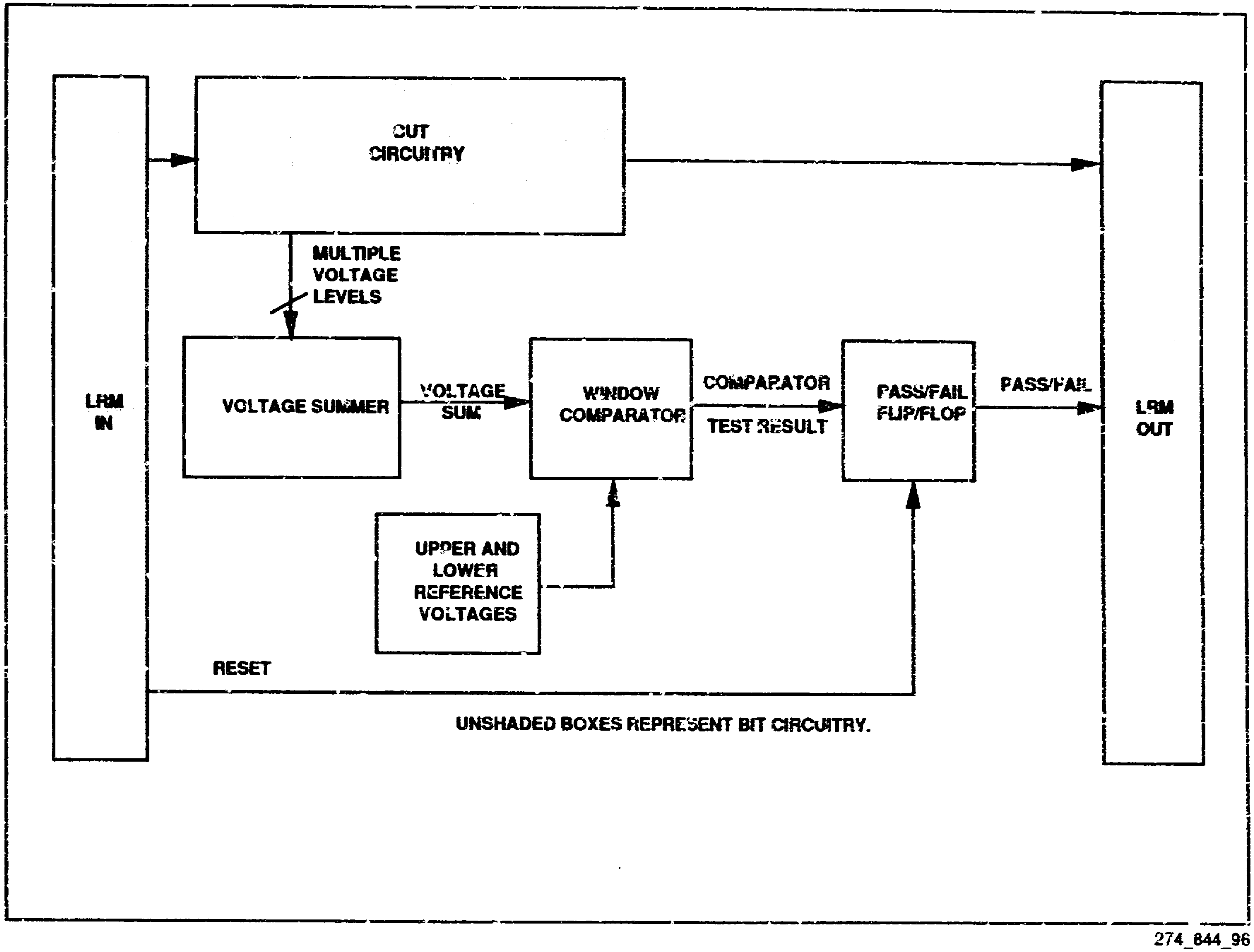


Figure 2 Level II Block Diagram For Voltage Summing Technique

**BIT TECHNIQUE: VOLTAGE SUMMING**

**CATEGORY: FLOW CHART / DESCRIPTION**

Figure 3 shows the Flow Chart for this BIT technique.

1. Circuit Under Test (CUT) is powered up.

2. The PASS/FAIL Flip-Flop is reset to PASS.

3. The voltage levels to be monitored are applied to a voltage summing circuit. This circuit adds the separate voltage levels and outputs the resulting sum.

4. The sum is then sent into a window comparator circuit which checks the signal against an upper and lower reference voltage.

5. If the sum is within the upper and lower reference voltages, the Pass/Fail Output remains low indicating a Pass condition. If the sum is out of tolerance, the Pass/Fail output signal presets to a high, indicating a failure condition.

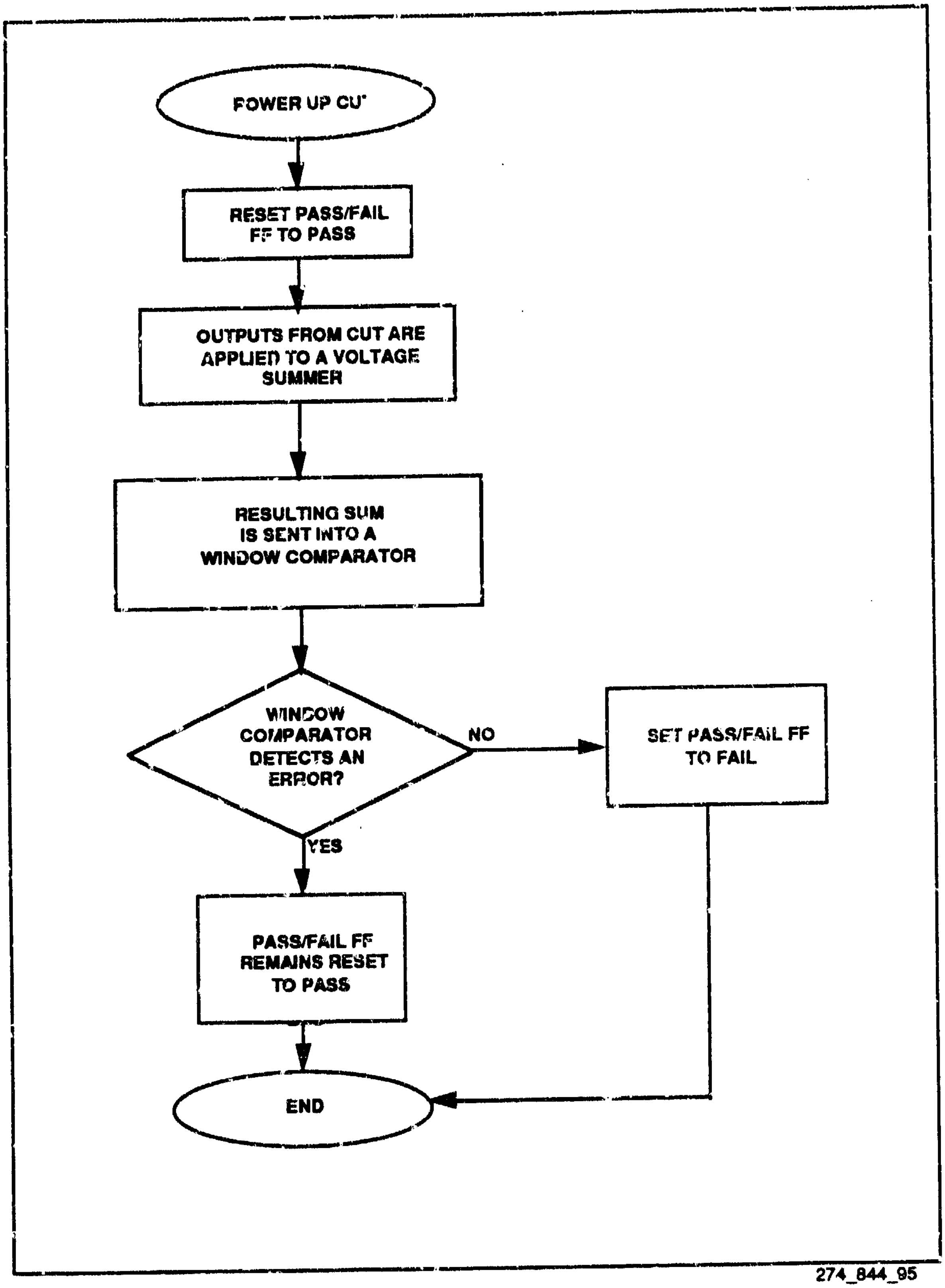

FOWER UP CU'
RESET PASS/FAIL FF TO PASS
OUTPUTS FROM CUT ARE APPLIED TO A VOLTAGE SUMMER
RESULTING SUM IS SENT INTO A WINDOW COMPARATOR
WINDOW COMPARATOR DETECTS AN ERROR?
NO
SET PASS/FAIL FF TO FAIL
YES
PASS/FAIL FF REMAINS RESET TO PASS
END
274_844_95

Figure 3 BIT Test Sequence Flow Chart For Voltage Summing BIT Technique

## BIT TECHNIQUE: VOLTAGE SUMMING

## CATEGORY: ADVANTAGES

Voltage Summing BIT has the following advantages to the circuit designer:

1. A minimum of components and board real estate are required to implement Voltage Summing compared to other techniques such as Comparator BIT.
2. Real estate saved increases with number of voltage level outputs monitored.
3. Voltage Summing BIT is a concurrent test, therefore end-to-end run time is not compromised. Failures are detected anytime during normal operation.
4. The basic component of a voltage summing circuit is the op amp, which possesses the following advantages:

The input impedance of an op amp is extremely high, minimizing Circuit Under Test (CUT) loading problems when incorporated into a BIT circuit.

Operational Amplifiers (Op Amps) are readily available "off the shelf" from a large number of manufacturers.

## CATEGORY: DISADVANTAGES

The Voltage Summing BIT technique has the following disadvantages:

1. The ability to verify the specification requirement of an individual voltage is reduced because only the sum of the voltage levels is monitored.
2. Reference voltages used in the window comparator must be provided and accurately maintained.
3. Voltage Summing BIT can only be used to monitor static signals.
4. The precision resistor required may have a long lead time.
5. Careful selection of the operational amplifiers is necessary to provide an accurate and stable result.
6. Greater precision is required as the number of channels increase. Worst case analysis will determine the practical maximum for each circuit.

## BIT TECHNIQUE: VOLTAGE SUMMING

## CATEGORY: ATTRIBUTES

1. CONCURRENCY

- The Voltage Summing BIT is a concurrent technique.

2. TECHNOLOGY

- Analog

3. CUT MICROPROCESSOR REQUIRED?

- No.

4 CUT INTERNAL DESIGN REQUIRED?

- No, only the normal outputs are monitored.

5. AREA PENALTY

- One op amp configured as a voltage summing amplifier.
- Two comparators configured as a window detector.
- One flip flop used as a latch.
- The number of resistors needed is directly proportional to the number of voltage levels to be summed, and may be calculated using the following equation:

Number of resistors = number of voltage levels to be summed, plus eight additional resistors for the default case.

6. WEIGHT PENALTY

- The nominal weight is equal to the weight of the summing amplifier, window detector, flip flop, and eight resistors.
- The weight increases as the number of voltage levels to be summed increases due to the addition of input resistors.

**BIT TECHNIQUE: VOLTAGE SUMMING**

**CATEGORY: ATTRIBUTES, Contd**

7. POWER PENALTY

- The nominal power dissipation is equal to the power dissipation of the summing amplifier, window detector, flip flop, and eight resistors.
- The power dissipation increases as the number of voltage levels to be summed increases due to the addition of input resistors.

8. TIMING PENALTY

- Since Voltage Summing BIT is done concurrently, there is no timing penalty.

9. RELIABILITY IMPACT

- The nominal failure rate of the summing amplifier, window comparator, flip flop, and eight resistors must be added to the total LRM failure rate.
- The failure rate increases proportionately as the number of voltage levels to be summed increases due to the addition of input resistors.

10. CONCEPTUAL COMPLEXITY

- Although the circuit design is relatively simple, a worst case analysis of the circuit is required to assure that false failures are not incurred.

11. HARDWARE / SOFTWARE / FIRMWARE

- Hardware
- For the summer amplifier, window comparator, flip flop, and resistors.

**BIT TECHNIQUE: VOLTAGE SUMMING**

**CATEGORY: ATTRIBUTES, Contd**

11. HARDWARE / SOFTWARE / FIRMWARE, Contd

- Software
- Not applicable.
- Firmware
- Not applicable.

12. DESIGN COST

- Hardware
- Proportional to the cost of the summer amplifier, window comparator, flip flop, and resistors.
- Software
- Not applicable.
- Firmware
- Not applicable.

13. MEMORY REQUIREMENTS

- Not applicable.

14. BIT CIRCUITRY SELF-TESTABLE ?

- No, but Voltage Summing can be made self testable with additional circuitry.

15 STAND-ALONE (SELF-CONTAINED ) BIT?

- Yes.

16. NOTES

- None.

## BIT TECHNIQUE: VOLTAGE SUMMING

## CATEGORY: DEFAULT DESIGN

Refer to Figure 4 for the Default Design schematic.

The voltage summing default design consists of voltage summing operational amplifier network, window comparator, and PASS/FAIL Flip-Flop. A reset input is provided to reset the PASS/FAIL Flip-Ficp Q output to a "low" (PASS) after the system has been powered for a sufficient time to overcome start-up transients.

The summing network is designed to provide a known good voltage level at the output of the summing amplifier when all CUT outputs applied to the summing amplifier are in specification. The summation of varying levels of CUT output voltages is accommodated by selection of the appropriate summing resistor values.

The summing amplifier adds the separate voltage levels and passes the results to a window comparator. If the summing amplifier output signal is greater than or less than the window limits, the comparator output voltage swings low and presets the PASS/FAIL Flip-Flop Q output to a high (FAIL). Otherwise, the Flip-Flop output remains at the Q=low (PASS) condition.

The designer, utilizing the BIT techniques, needs to determine the maximum allowable variation of summing amplifier output based on specification limits of the CUT outputs to be tested and the tolerance shifts of BIT circuitry. A worst case analysis is required and includes the calculation of the following resistor values:

| | |
|---|---|
| R1 | Summing network resistor for CUT output 1. |
| R2 | Summing network resistor for CUT output 2. |
| R3 | Summing network resistor for CUT output 3. |
| $Rfb$ | Sets the gain of the Summing Amplifier. |
| $R0ff$ | Selected to minimize output offsets due to input bias currents. |
| Ra | Applicable to window comparator limits |
| Rb | Applicable to window comparator limits |
| Rc | Applicable to window comparator limits |
| Rd | Applicable to window comparator limits |

**BIT TECHNIQUE: VOLTAGE SUMMING**

**CATEGORY: DEFAULT DESIGN, Contd**

BASIC DESIGN EQUATIONS

Summing Amplifier Equations:

$$V_{OUT} = -Rfb\left(\frac{V1}{R1} + \frac{V2}{R2} + \cdots \frac{Vn}{Rn}\right) \quad [1]$$

$$R_{OFF} = \frac{R1 \times R2 \times \ldots\ Rn \times Rfb}{R1 + R2 + \ldots\ Rn + Rfb} \quad [2]$$

Where:

| | | |
|---|---|---|
| V1 thru Vn | = | CUT output voltages 1 thru n |
| R1 thru Rn | = | Summing resistor associated with CUT output voltages 1 thru n |

Comparator Equations:

$$V_{LTP} = \frac{(Vcc - Vee)Rd}{Rc+Rd} + Vee \quad [3]$$

$$V_{UTP} = \frac{(Vcc - Vee)Rb}{Ra+Rb} + Vee \quad [4]$$

Where:

| | |
|---|---|
| $V_{LTP}$ = | Voltage (lower trigger point) |
| $V_{UTP}$ = | Voltage (upper trigger point) |
| $V_{REF\_POS}$ = | Positive reference voltage |
| $V_{REF\_NEG}$ = | Negative reference voltage |

Limitations:

- This BIT technique places a loading path between individual CUT outputs. The CUT outputs must be evaluated for compatibility with this technique.
- The summing amplifier output voltages must be limited to prevent exceeding the comparator differential input voltage range (±5V) per LM119.
- As the number of CUT outputs are summed, the accuracy requirements increase for the BIT summing amplifier and window comparator. The ability to detect an out of tolerance condition on each individual output, therefore, is reduced.
- Equal weighting factors for all CUT outputs may not be desirable for the summing network in some cases. For example, equal weighting applied to the outputs of a tracking ±15V power supply will not allow fault detection using this BIT technique. In this case, a change to apply unequal weighting will suffice.

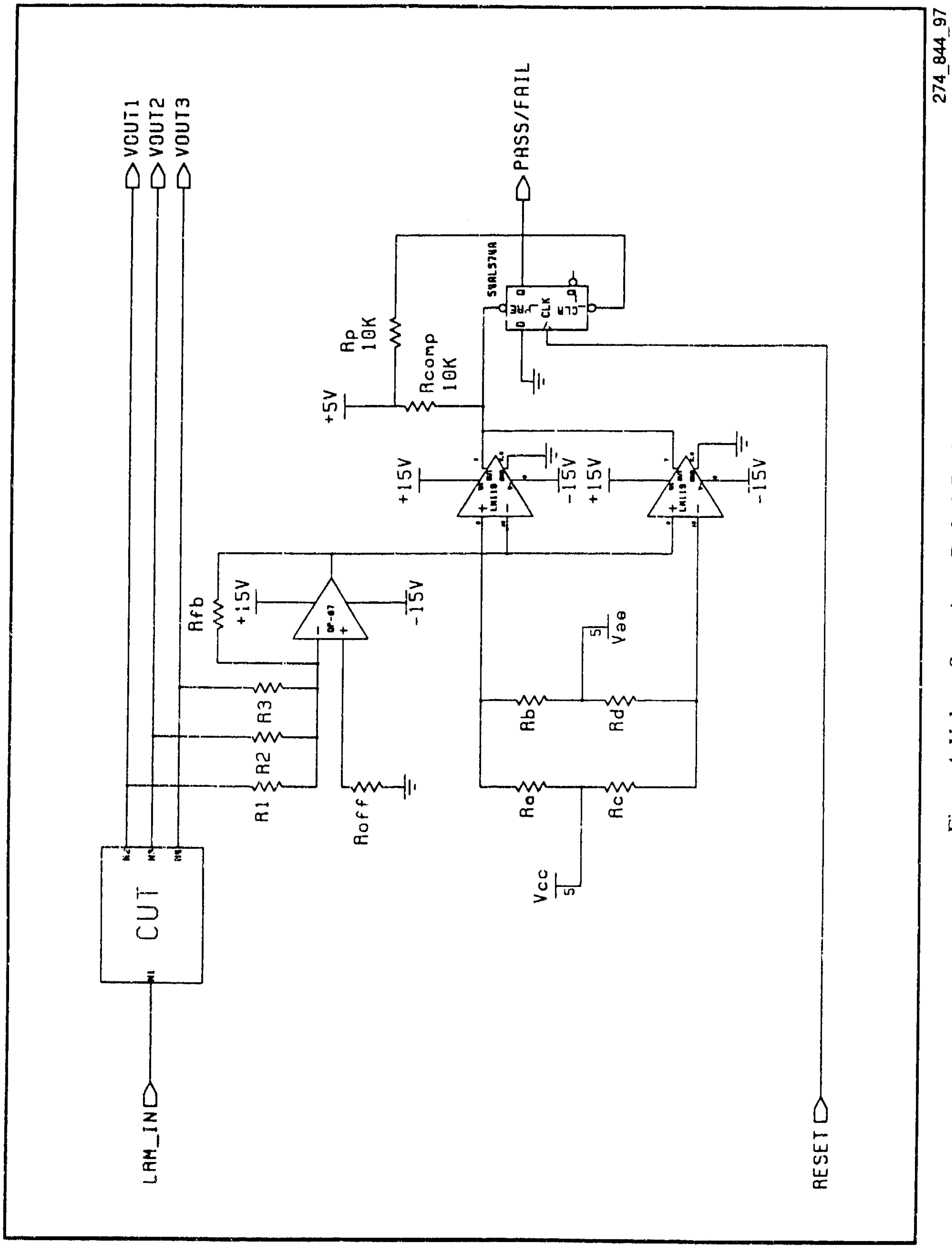


Figure 4 Voltage Summing – Default Design

## BIT TECHNIQUE: VOLTAGE SUMMING

## CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM

Figure 5 shows the BIT Technique Insertion Diagram for this BIT technique.

BIT ELEMENT 1

1. Connect the CUT analog outputs to the summing resistors R1 thru R3 per the default design figure 4.
2. Connect the summing resistors to the summing amplifier negative input per the default design figure 4.
3. Connect resistors Ro*ff* and Rfb, power, and ground to the summing amplifier per the default design figure 4.

BIT ELEMENT 2

1. Connect two LM119 comparators as a window comparator per the default design figure 4.
2. Connect the 0P-07 amplifier output to the window comparator input per the default design figure 4.
3. Connect resistors, power, and ground per the default design figure 4.

BIT ELEMENT 3

1. Connect the window comparator output to the PASS/FAIL flip-flop per the default design figure 4.
2. Connect the reset input to the flip-flop.
3. Connect the PASS/FAIL output to the LRM out.

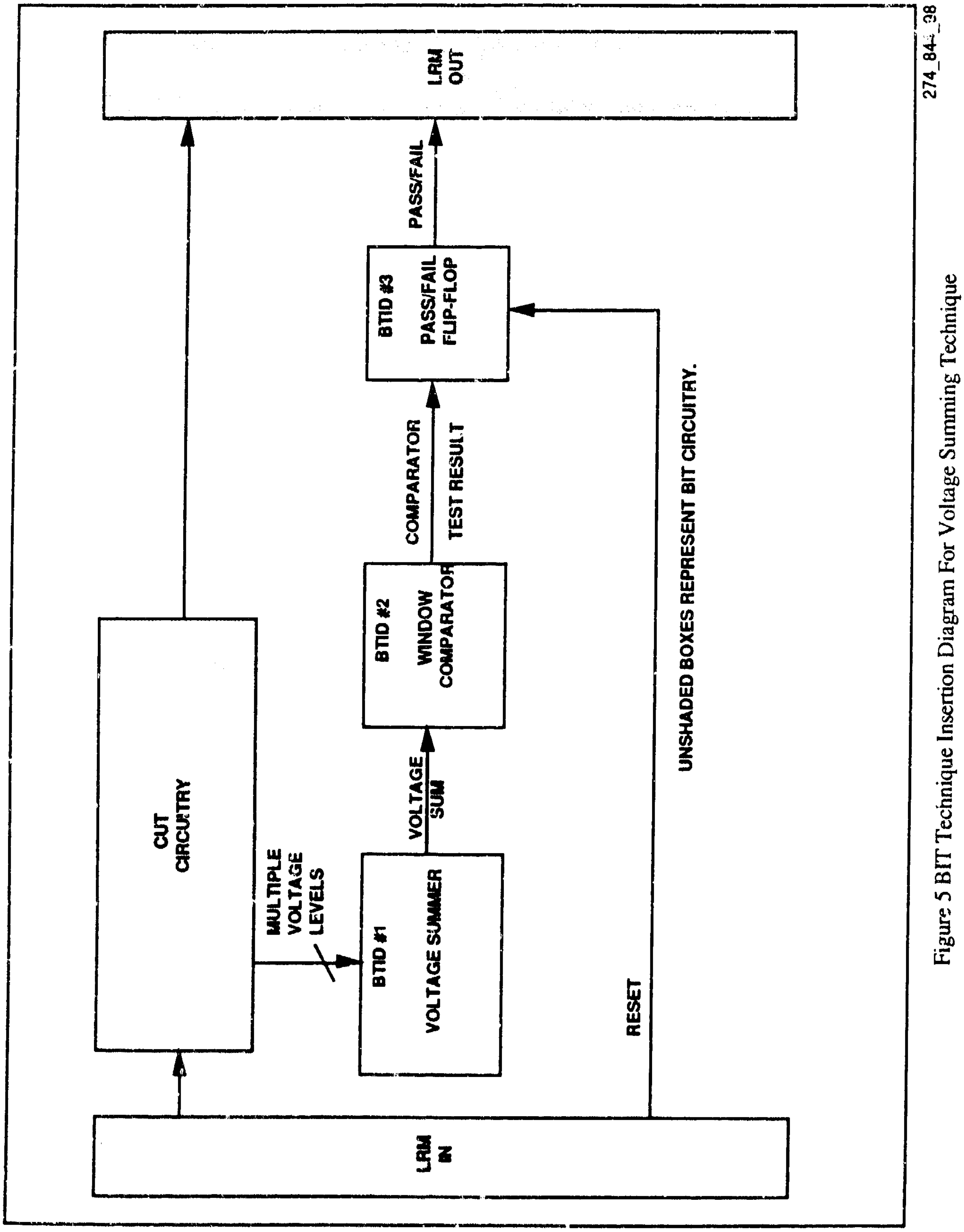


Figure 5 BIT Technique Insertion Diagram For Voltage Summing Technique

## BIT TECHNIQUE: VOLTAGE SUMMING

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | Number of CUT outputs to test | none |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 54ALS74A | FLIP-FLOP | 1 |
| 2 | LM119 | SELECTOR/MUX | 2 |
| 3 | OP-07 | COUNTER | 1 |
| 4 | RNC55 | BUFFER | 2 |
| 5 | RNC60 | COMPARATOR | $v_1 + 6$ |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i = \text{ceil}\,(u_i / upp_i)$$

**Explanation of symbols used:**

| | | |
|---|---|---|
| $n_i$ | = | Number of components (physical packages) required for i-th part |
| $u_i$ | = | Number of units (CAD symbols) required for i-th part |
| $upp_i$ | = | Number of units/package for i-th part (from Table 4.0) |
| ceil | = | Round up to nearest integer (e.g. 5.3 is rounded to 6) |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |

## BIT TECHNIQUE: VOLTAGE SUMMING

## CATEGORY: PENALTY EQUATIONS

**AREA (sq.in.)** = Sum ( $n_i * a_i$ ) + 15% for traces

**WEIGHT (gms)** = Sum ( $n_i * w_i$ ) + 10% for solder

**POWER (mW)** = Sum ( $n_i * p_i$ )

**TEST TIME (ns)** = 0

**DELAY (ns)** = 0

**Explanation of Symbols Used.**

| | | |
|---|---|---|
| Sum | = | Sum over parts in BIT Technique Parts Table (i=1 to 5) |
| $n_i$ | = | Number of component packages for i-th part in Parts Table<br>$n_i$ is calculated according to Component Determination Equations |
| $a_i$ | = | Area in sq.in. of i-th part in Parts Table |
| $p_i$ | = | Power dissipation in mW (typical) for i-th part in Parts Table |
| $w_i$ | = | Weight in grams of i-th part in Parts Table |

## BIT TECHNIQUE: VOLTAGE SUMMING

## CATEGORY: BIBLIOGRAPHY

None required.

# REDUNDANCY

# BIT TECHNIQUE DATA PACKAGE

## CATEGORY: SHORT DESCRIPTION / DIAGRAMS

Redundant test techniques can be implemented on a concurrent basis by the design engineer as a Built-In-Test (BIT) tool. The approach taken is to include a Standard (also known as a golden device) on the Line Replaceable Module (LRM). The Standard is an electrical replica of the Circuit Under Test (CUT) that requires BIT capability. The outputs from the CUT and the Standard are fed into a differential amplifier circuit which in turn feeds its output into a window comparator. The window comparator generates a FAIL signal if the differential output signal is either above a positive reference level or below a negative reference level.

Figures 1 and 2 show the Level I and II Block Diagrams for this BIT technique.

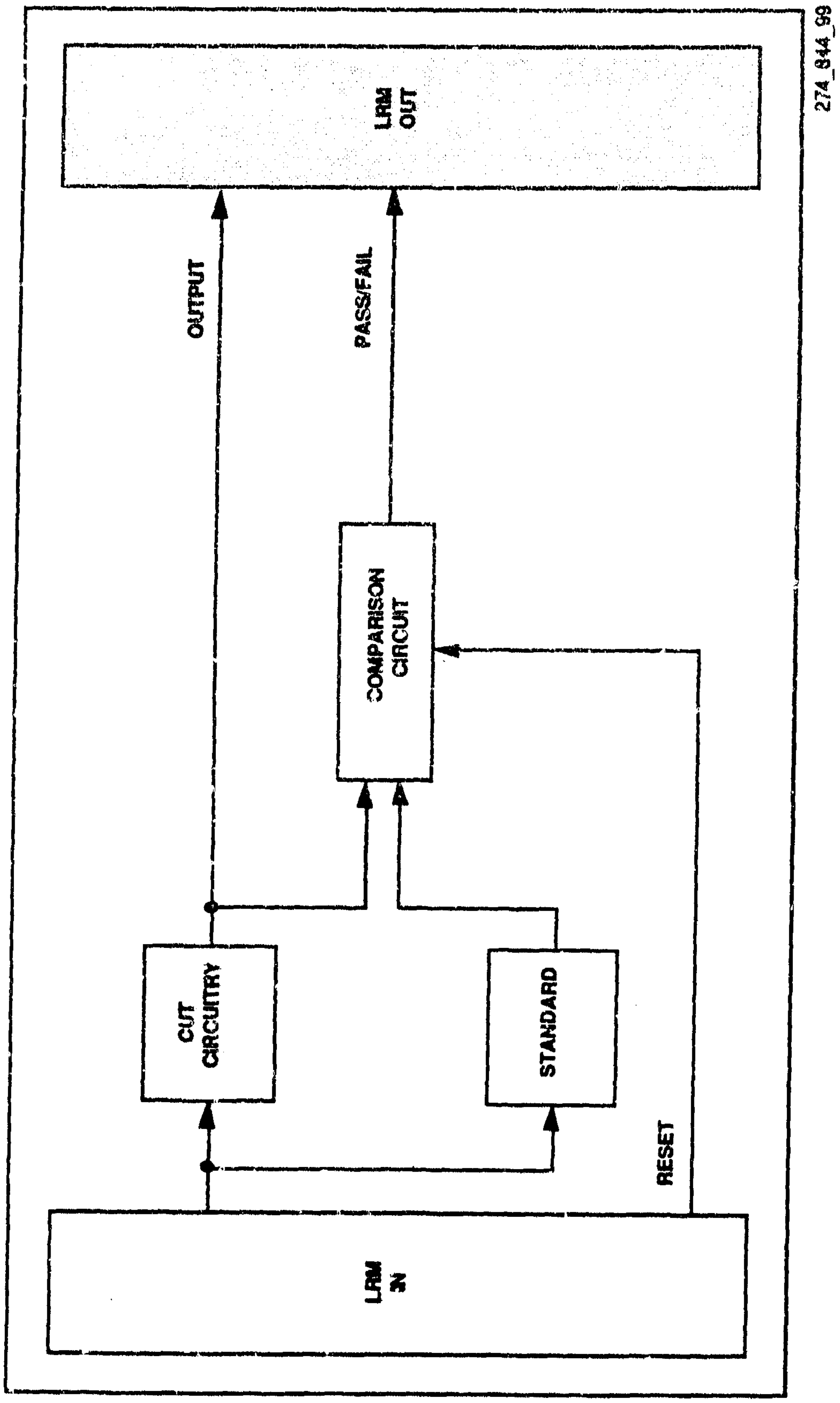
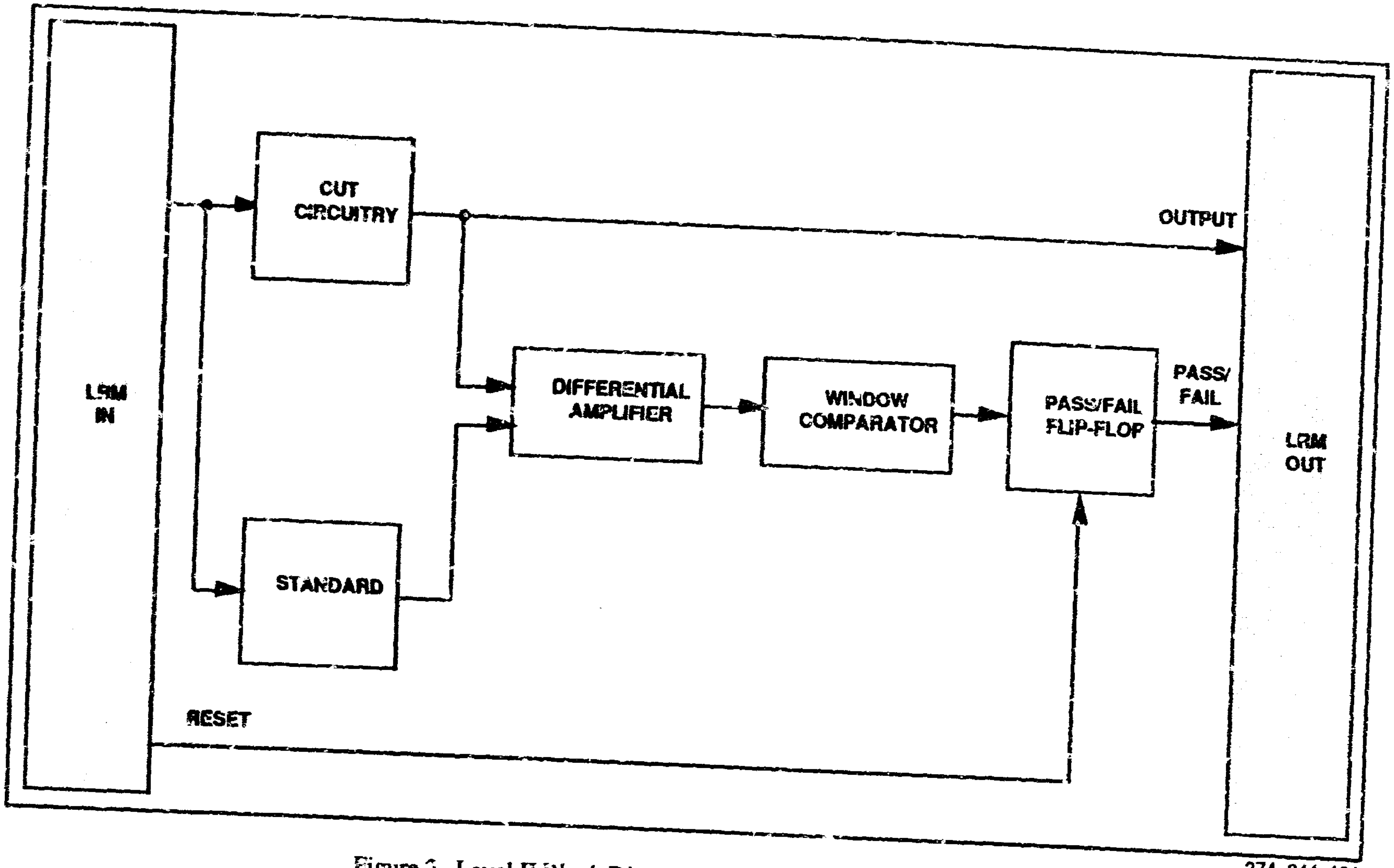


Figure 1 Level I Block Diagram For Redundancy BIT Technique



274_844_101

Figure 2 Level II Block Diagram for Redundancy BIT Technique

## BIT TECHNIQUE: REDUNDANCY

## CATEGORY: FLOW CHART / DESCRIPTION

Figure 3 shows the Flow Chart for this BIT technique.

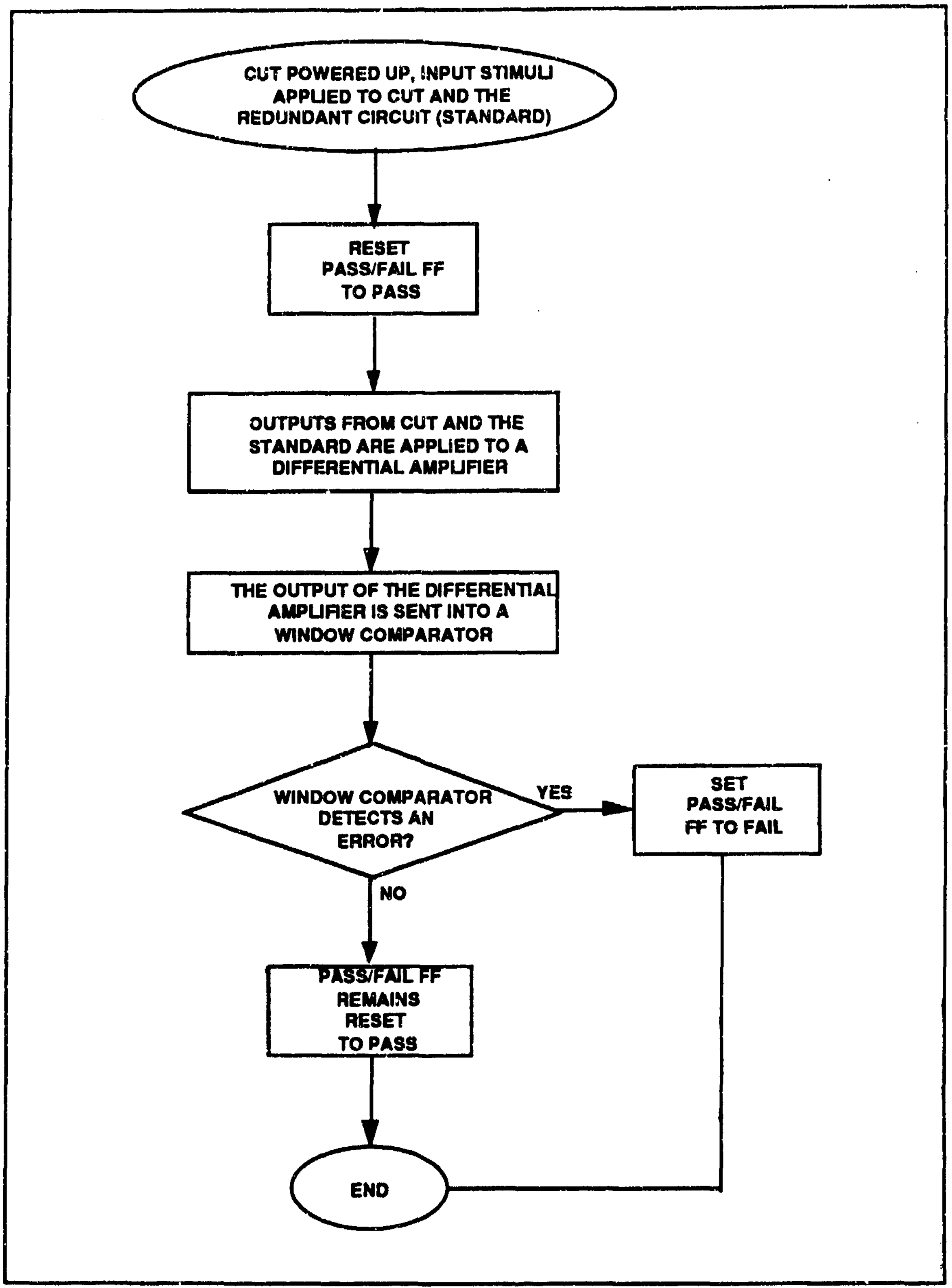
1. Signals sent to the CUT are sent concurrently to a replica of the CUT (the Standard) during operation of the LRM.

2. The PASS/FAIL Flip-Flop is reset to PASS.

3. Both the CUT output and the output from the Standard are sent to a differential amplifier which provides an output which is proportional to the difference between the two signals.

4. The output of the differential amplifier is sent to the inputs of a window comparator where they are compared to positive and negative reference voltages.

5. If the output of the differential amplifier is more positive or more negative than the reference voltages, the PASS/FAIL Flip-Flop is set to FAIL. Otherwise, the Flip-Flop remains reset.


CUT POWERED UP, INPUT STIMULI APPLIED TO CUT AND THE REDUNDANT CIRCUIT (STANDARD)
RESET PASS/FAIL FF TO PASS
OUTPUTS FROM CUT AND THE STANDARD ARE APPLIED TO A DIFFERENTIAL AMPLIFIER
THE OUTPUT OF THE DIFFERENTIAL AMPLIFIER IS SENT INTO A WINDOW COMPARATOR
WINDOW COMPARATOR DETECTS AN ERROR?
YES
SET PASS/FAIL FF TO FAIL
NO
PASS/FAIL FF REMAINS RESET TO PASS
END

274_844_100

Figure 3 BIT Sequence Flow Chart For Redundancy BIT Technique

## BIT TECHNIQUE: REDUNDANCY

### CATEGORY: ADVANTAGES

1. The technique operates concurrently with normal operation, therefore, no time is lost to test of the CUT.

2. All the circuit design must have been previously done for the CUT; therefore, it is readily available for the Standard.

### CATEGORY: DISADVANTAGES

1. In high frequency or critical timing applications, it may be difficult to synchronize the CUT output signals with those of the redundant circuit.

2. Transient output changes of the CUT and the Standard may be difficult to test due to common mode rejection and bandwidth limitations of the differential amplifier.

3. Large amounts of real estate will be consumed because the technique may double the circuit area required.

4. For a large number of CUT outputs, many differential amplifier and comparator BIT circuits would be required.

**BIT TECHNIQUE: REDUNDANCY**

**CATEGORY: ATTRIBUTES**

1. CONCURRENCY

   - Test is concurrent with operational use.

2. TECHNOLOGY

   - Applicable to CUT circuits using analog technology.

3. CUT MICROPROCESSOR REQUIRED?

   - No.

4. CUT INTERNAL DESIGN REQUIRED?

   - No.

5. AREA PENALTY

   - The increase in area is proportional to the area of the CUT which is replicated plus the area of the differential amplifier, window comparator, and PASS/FAIL Flip-Flop.

6. 'VEIGHT PENALTY

   - The increase in weight is proportional to the weight of the CUT which is replicated plus the weight of the differential amplifier, window comparator, and PASS/FAIL Flip-Flop.

7. POWER PENALTY

   - The increase in power is proportional to the power of the CUT which is replicated plus the power of the differential amplifier, window comparator, and PASS/FAIL Flip-Flop.

## BIT TECHNIQUE: REDUNDANCY

## CATEGORY: ATTRIBUTES, Contd

8. TIMING PENALTIES

- Test Time
  - The test time is negligible: It is equal to the sum of the delays due to the differential amplifier, window comparator, and the PASS/FAIL Flip-Flop.
  - If high frequency or critical timing specifications are required by the CUT, then difficulties may arise when attempting to synchronize the CUT's output with the output of the Standard.
- Throughput Delay
  - No timing penalty is incurred with the use of this technique since it executes concurrently with normal operation.

9. RELIABILITY IMPACT

- With the use of this technique, failure rates increase proportionately to the circuit density and complexity of the CUT. The failure rates for the differential amplifier, comparator circuit, and PASS/FAIL Flip-Flop must also be added to the total LRM failure rate.

10. CONCEPTUAL COMPLEXITY

- Straightforward

11. HARDWARE/SOFTWARE/FIRMWARE

- The hardware implementation requires a differential amplifier, window comparator, PASS/FAIL Flip-Flop, and a Standard. The Standard is an electrical replica of the CUT.
- Software and Firmware: Not applicable

12. DESIGN COST

- Hardware: Proportional to the cost of the Standard plus the differential amplifier, window comparator, and PASS/FAIL Flip-Flop.
- Software and Firmware: Not applicable

## BIT TECHN!QUE: REDUNDANCY

## CATEGORY: ATTRIBUTES, Contd

13. MEMORY REQUIREMENTS

- Not applicable

14. BIT CIRCUITRY SELF-TESTABLE?

- Since the CUT is duplicated by the BIT circuitry, a failure in either the CUT or BIT circuitry is detectable, therefore, the majority of the circuitry added for a BIT is checked out. Only the differential amplifier and window comparator circuits are not verified operationally.

15. STAND-ALONE (SELF-CONTAINED) BIT?

- Yes.

16. NOTES

- None.

## BIT TECHNIQUE: REDUNDANCY

## CATEGORY: DEFAULT DESIGN

Refer to Figure 4 for the Default Design schematic.

The Redundancy BIT technique consists of a differential amplifier, window comparator, PASS/FAIL Flip-Flop, and a Standard; the latter is an electrical replica of the CUT. A reset input is provided to reset the PASS/FAIL Flip-Flop output to a low (PASS) after the system has been powered for a sufficient time to overcome start-up transients.

The analog output signals from the CUT and Standard, which ideally are identical, are passed to a differential amplifier. The differential amplifier amplifies the difference between the two signals and passes the result to the window comparator. If the difference signal is greater than or less than the window limits, the comparator output voltage swings low and presets the PASS/FAIL Flip-Flop output to a high (fail). Otherwise, the Flip-Flop output remains at the low (PASS) condition.

The designer utilizing this BIT technique must determine the maximum allowable error between the CUT and the Standard, and then calculate the necessary resistor values listed below:

| | |
|---|---|
| R1 | Applicable to window comparator limits |
| R2 | Applicable to window comparator limits |
| R3 | Applicable to window comparator limits |
| R4 | Sets the gain of the differential amplifier |

**BIT TECHNIQUE: REDUNDANCY**

**CATEGORY: DEFAULT DESIGN, Contd**

BASIC DESIGN EQUATIONS

Differential Amplifier Equations:

$$V_O = A_D\,[(V_{CUT} - V_{STD} + V_{io}) + \frac{1}{CMRR}(V_{CM})] + V_{OO} \qquad [1]$$

Where:

| | | |
|---|---|---|
| $V_O$ | = | Output voltage of the differential amplifier (DA) |
| $A_D$ | = | Differential gain of the DA |
| $V_{CUT}$ | = | Output voltage of the CUT |
| $V_{STD}$ | = | Output voltage of the Standard |
| $V_{io}$ | = | Input offset voltage of the DA |
| $V_{oo}$ | = | Output offset voltage of the DA |
| CMRR | = | Common Mode Rejection Ratio of the DA |
| $V_{CM}$ | = | Common Mode input voltage of the DA |

$$R_4 = \frac{40.000}{G - 1} \qquad [2]$$

Where:

| | | |
|---|---|---|
| G | = | Differential gain of the DA |
| $R_4$ | = | Feedback resistor in default design |

**BIT TECHNIQUE: REDUNDANCY**

**CATEGORY: DEFAULT DESIGN, Contd**

Comparator Equations:

$$V_{LTP} = \frac{[(VREF\_POS) - (VREF\_NEG)]R3}{R1 + R2 + R3} + VREF\_NEG \qquad [3]$$

$$V_{UTP} = \frac{[(VREF\_POS) - (VREF\_NEG)](R2+R3)}{R1 + R2 + R3} + VREF\_NEG \qquad [4]$$

$V_{LTP}$ = Voltage (lower trigger point)

$V_{UTP}$ = Voltage (upper trigger point)

Limitations

- Transient Response–The rate of change in the analog signals of the CUT and the Standard must be considered in the BIT circuit design, e.g. a wider window limit may be necessary due to the common mode rejection and bandwidth limitations of the differential amplifier. A worst case analysis is necessary.
- This BIT technique adds loading to the CUT output, i.e. 100KΩ.
- The differential amplifier output voltages must be limited to prevent exceeding the comparator differential input voltage range (±5V) per LM119.

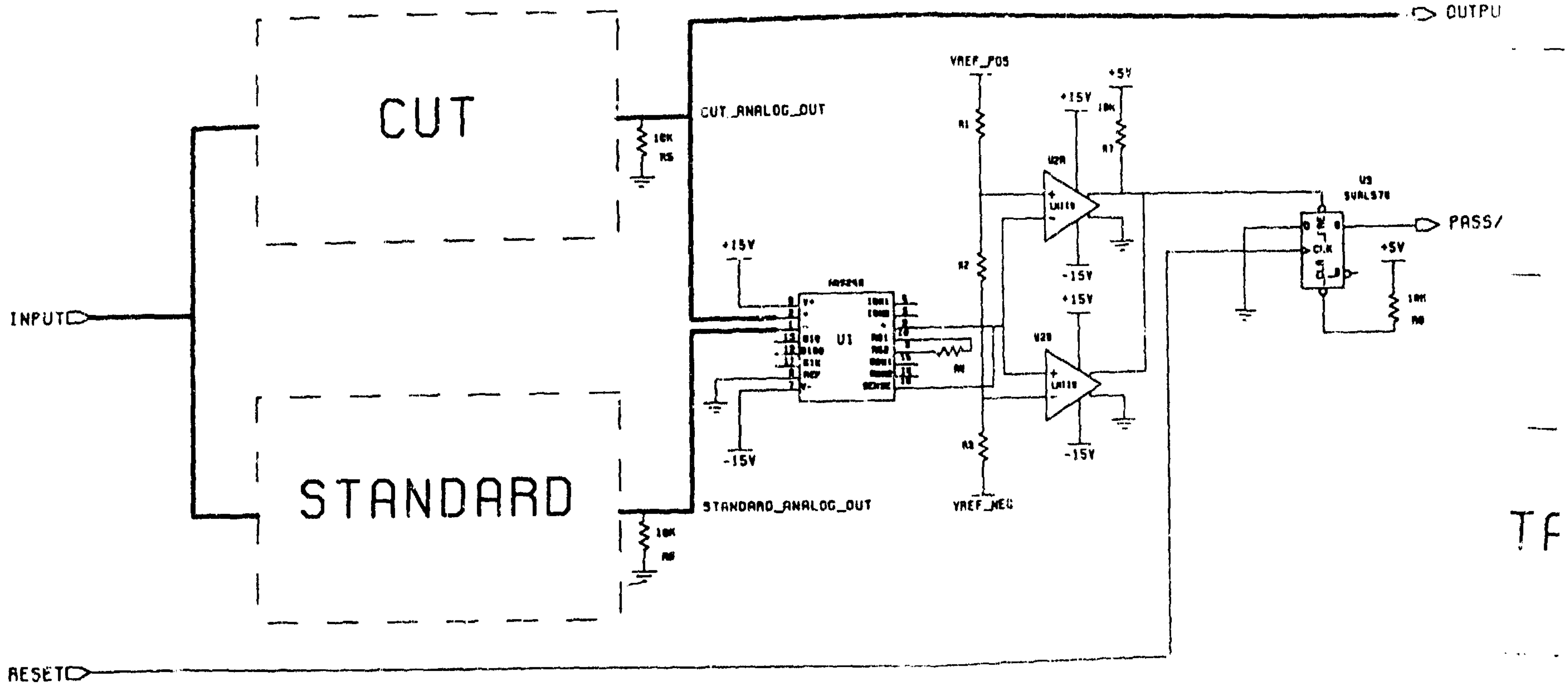


Figure 4 Redundancy – Default Design

1

2

Re

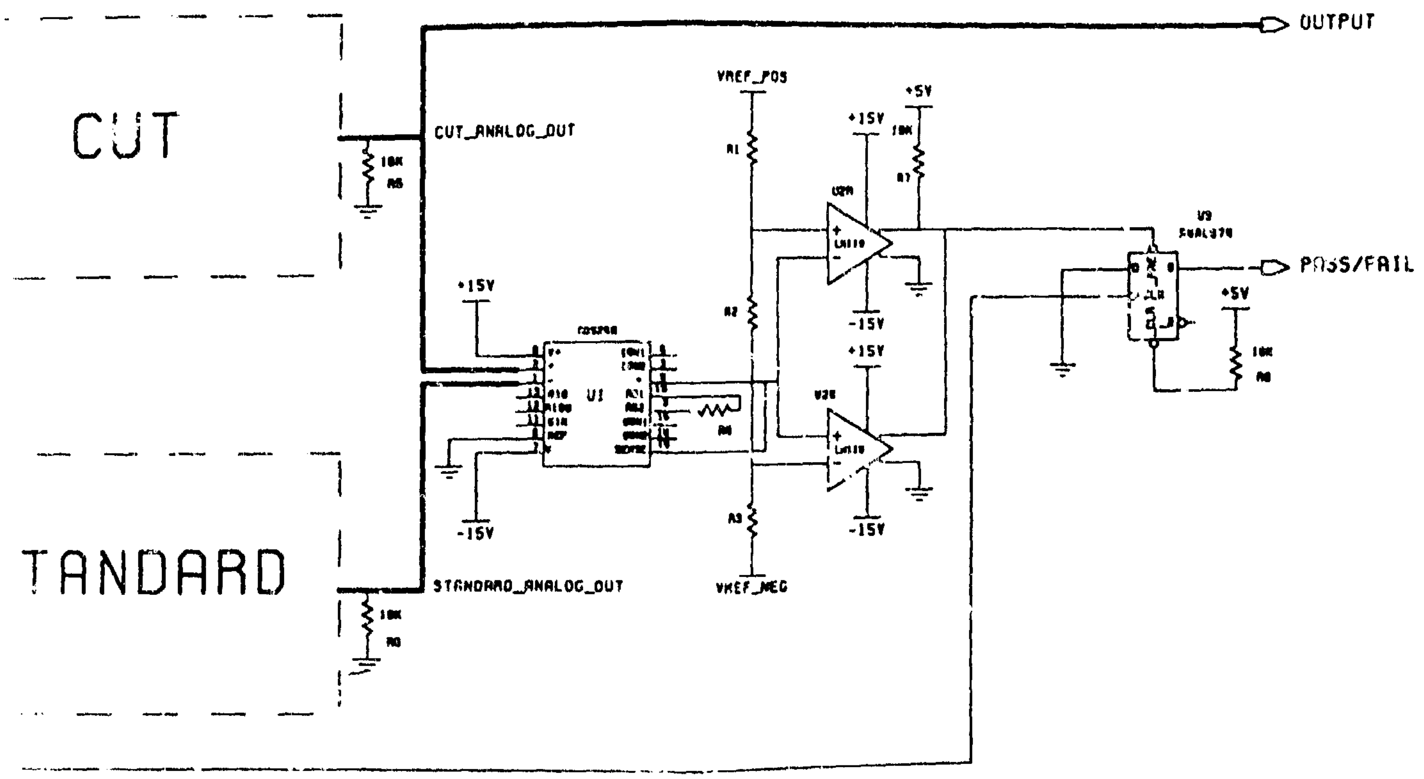


Figure 4 Redundancy – Default Design

**BIT TECHNIQUE: REDUNDANCY**

**CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM**

Figure 5 shows the BIT Technique Insertin Diagram for this BIT technique.

BIT ELEMENT 1

1. Connect the CUT analog output to the AD524 differential amplifier positive input per the default design figure 4.

2. Connect the Standard analog output to the AD524 differential amplifier negative input per the default design figure 4.

3. Connect resistors, power, jumper, and ground to the AD524 per the default design figure 4.

BIT ELEMENT 2

1. Connect two LM119 comparators as a window comparator per the default design figure 4.

2. Connect the AD524 differential amplifier output to the window comparator input per the default design figure 4.

3. Connect resistors, power, jumper, and ground per the default design figure 4.

BIT ELEMENT 3

1. Connect the window comparator output to the PASS/FAIL flip-flop per the default design figure 4.

2. Connect the reset input to the flip-flop.

3. Connect the PASS/FAIL output to the LRM out.

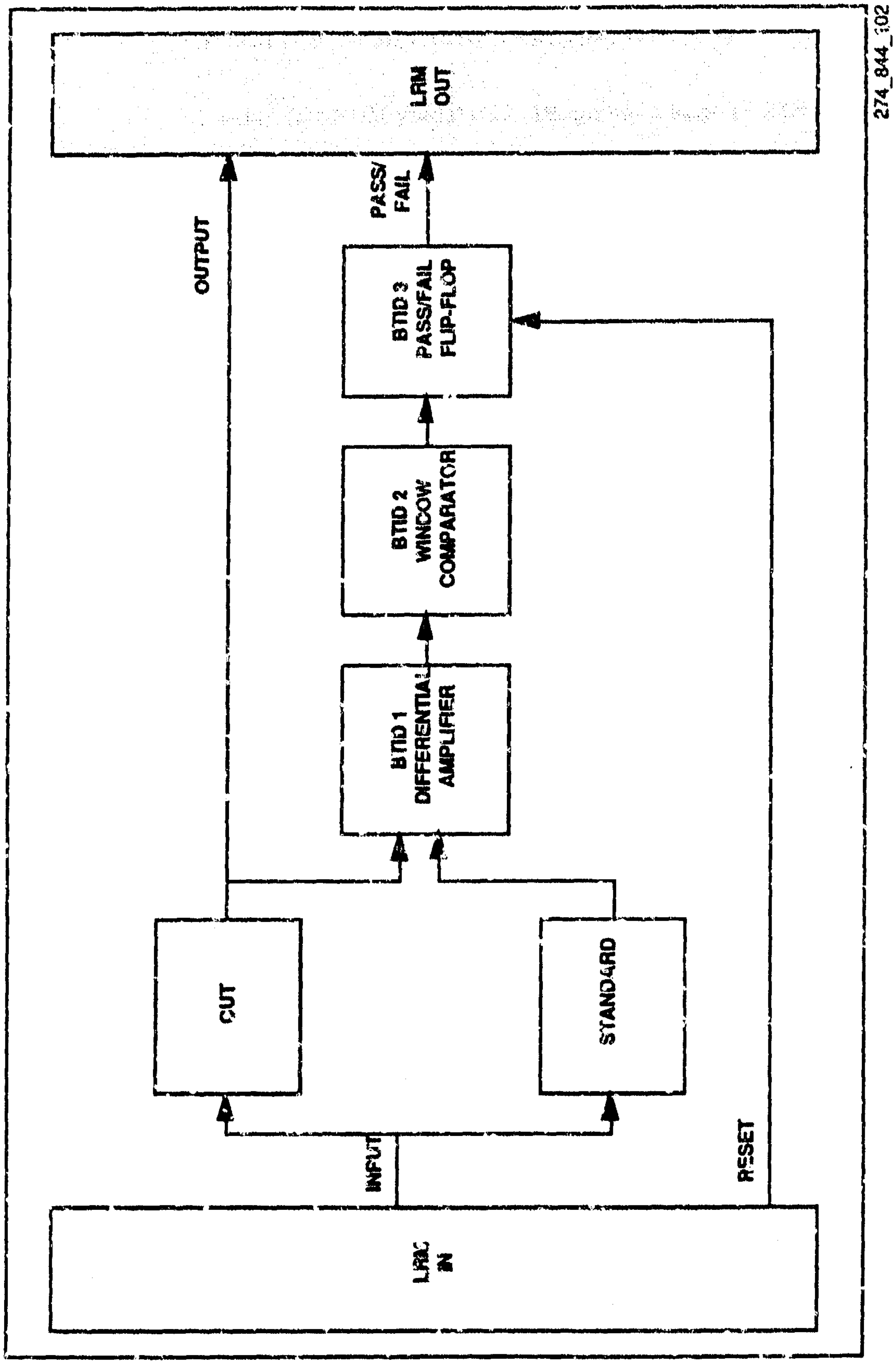


Figure 5 BIT Technique Insertion Diagram for Redundancy BIT Technique

## BIT TECHNIQUE: REDUNDANCY

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | Number of CUT outputs to test | none |
| $v_2$. | Board area of Standard for Redundancy BIT | sq.in. |
| $v_3$. | Power (typical) of Standard for Redundancy BIT | mW |
| $v_4$. | Weight of Standard for Redundancy BIT | gms |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | 54ALS74A | FLIP-FLOP | $v_1$ |
| 2 | LM119 | COMPARATOR | $2v_1$ |
| 3 | AD524A | DIFF AMPLIFIER | $v_1$ |
| 4 | RNR55 | RESISTOR | $8v_1$ |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i = \text{ceil}\,( u_i / upp_i )$$

Explanation of symbols used:

$n_i$ = Number of components (physical packages) required for i-th part<br>
$u_i$ = Number of units (CAD symbols) required for i-th part<br>
$upp_i$ = Number of units/package for i-th part (from Table 4.0)<br>
ceil = Round up to nearest integer (e.g. 5.3 is rounded to 6)<br>
$v_i$ = User-supplied value for i-th variable (see Variable Definitions)

## BIT TECHNIQUE: REDUNDANCY

## CATEGORY: PENALTY EQUATIONS

**AREA (sq.in.)** = Sum ( $n_i * a_i$ ) + 15% for traces

**WEIGHT (gms)** = Sum ( $n_i * w_i$ ) + 10% for solder

**POWER (mW)** = Sum ( $n_i * p_i$ )

**TEST TIME (ns)** = 0

**DELAY (ns)** = 0

**Explanation of Symbols Used.**

| | | |
|---|---|---|
| Sum | = | Sum over parts in BIT Technique Parts Table (i=1 to 4) |
| $n_i$ | = | Number of component packages for i-th part in Parts Table<br>$n_i$ is calculated according to Component Determination Equations |
| $a_i$ | = | Area in sq.in. of i-th part in Parts Table |
| $p_i$ | = | Power dissipation in mW (typical) for i-th part in Parts Table |
| $w_i$ | = | Weight in grams of i-th part in Parts Table |

## CATEGORY: BIBLIOGRAPHY

None required.

# ANALOG WRAPAROUND

# BIT TECHNIQUE DATA PACKAGE

## CATEGORY: SHORT DESCRIPTION / DIAGRAMS

Analog Wraparound is a non-concurrent Built-In-Test (BIT) Technique for systems employing an analog input and output control structure, e.g. servo controllers, autopilot transceivers, and two-way communication links.

The Analog Wraparound technique consists of hardware and firmware in Read Only Memory [ROM] and specifically requires a microprocessor, some Digital-to-Analog (D/A) converters as output devices and some Analog-to-Digital (A/D) converters as input devices on board as part of the Circuit Under Test (CUT).

The technique consists of adding necessary circuitry so that upon a BIT INITIATE, the analog signal leaving the D/A output devices can be routed to the A/D input devices on the Line Replaceable Module (LRM). An appropriate BIT routine is stored in ROM along with test data to control the data transfer and compare the data received with the data transmitted. A mismatch will indicate a failure.

An option which the engineer may use to route the signal back to the microprocessor is to add an analog switch to wrap the signal leaving the D/A through the switch, then to the A/D, and back to a microprocessor.

Any analog command/response pair can be tested using the Analog Wraparound method.

The Microprocessor BIT technique (a related BIT technique) checks out the internal components of the microprocessor system. The analog wraparound BIT technique can be used to extend the Microprocessor BIT to include the I/O.

Figures 1 and 2 show the Level I and II Block Diagrams for this BIT technique.

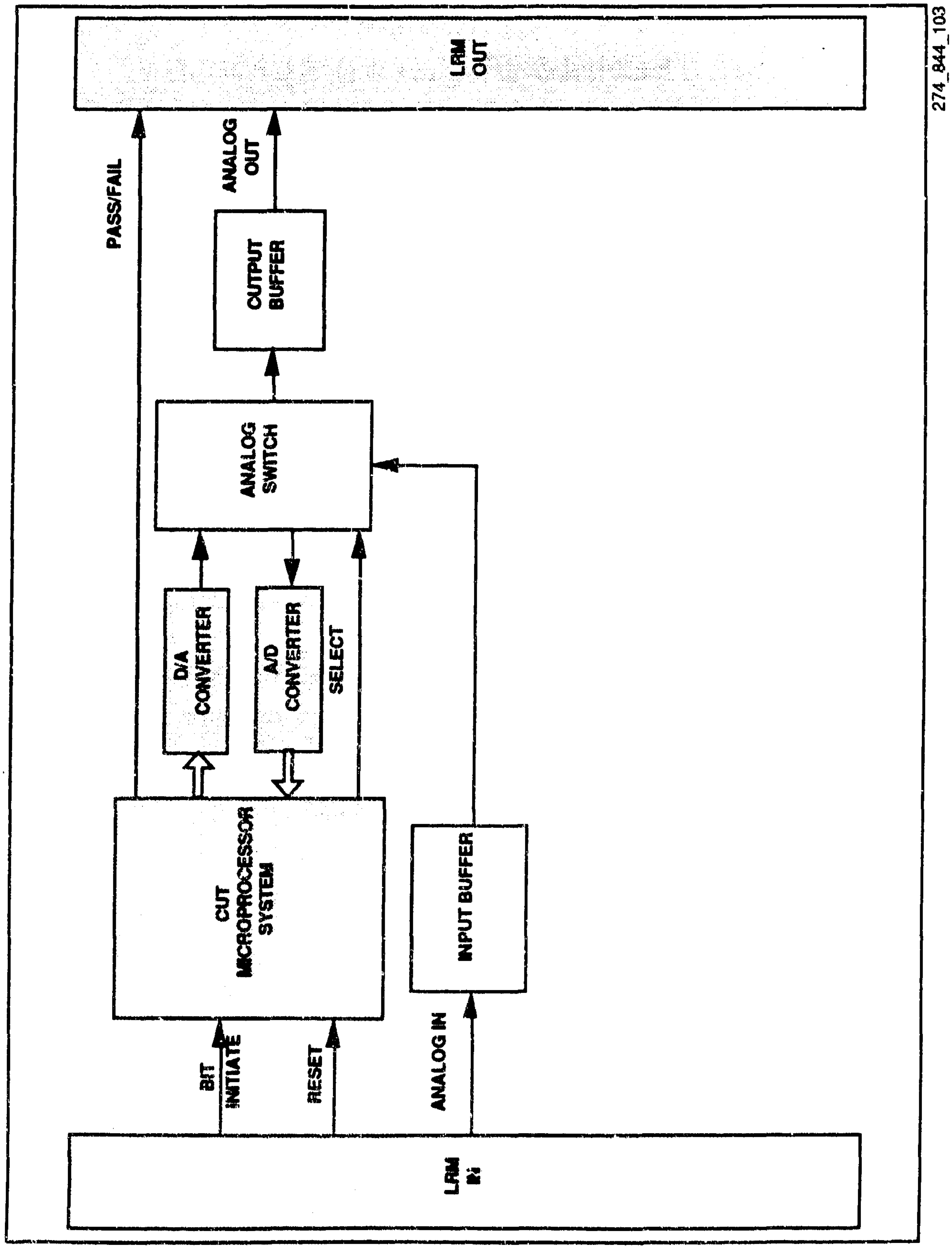


Figure 1 Level 1 Block Diagram Utilizing Analog Wraparound

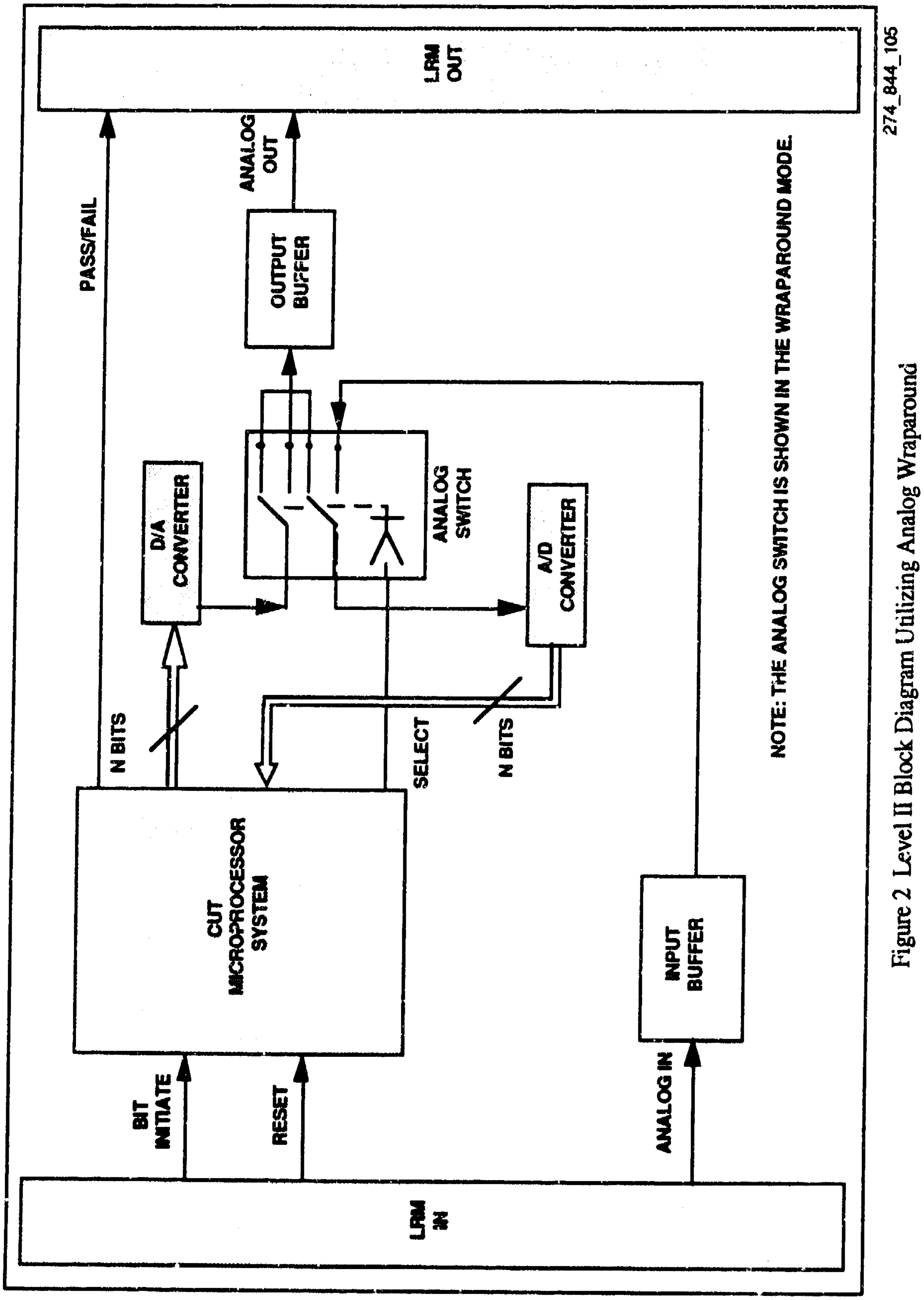


Figure 2 Level II Block Diagram Utilizing Analog Wraparound

## BIT TECHNIQUE: ANALOG WRAPAROUND

## CATEGORY: FLOW CHART DESCRIPTION

Figure 3 shows the Flow Chart for this BIT technique.

1. A 'BIT INITIATE' signal is input from the LRM, so testing can begin.
2. Initialize the Circuit Under Test and set the Pass/Fail indicator equal to PASS.
3. Before applying a signal, enable the wraparound switches that are going to be used for that particular test.
4. Apply the ROM test patterns to the D/A converter(s).
5. At this point, the data is routed from the D/A converter through the proper enabled wraparound switches and into the A/D converter(s).
6. Delay and strobe the memory chip to send the expected results to the microprocessor.
7. Microprocessor reads the results from the A/D converter(s) and compares it with the expected result from memory.
8. If comparison fails, set Pass/Fail to FAIL and end test. If comparison passes, continue.
9. If not the last ROM address, apply the next ROM test pattern and go back to STEP 5 and continue.

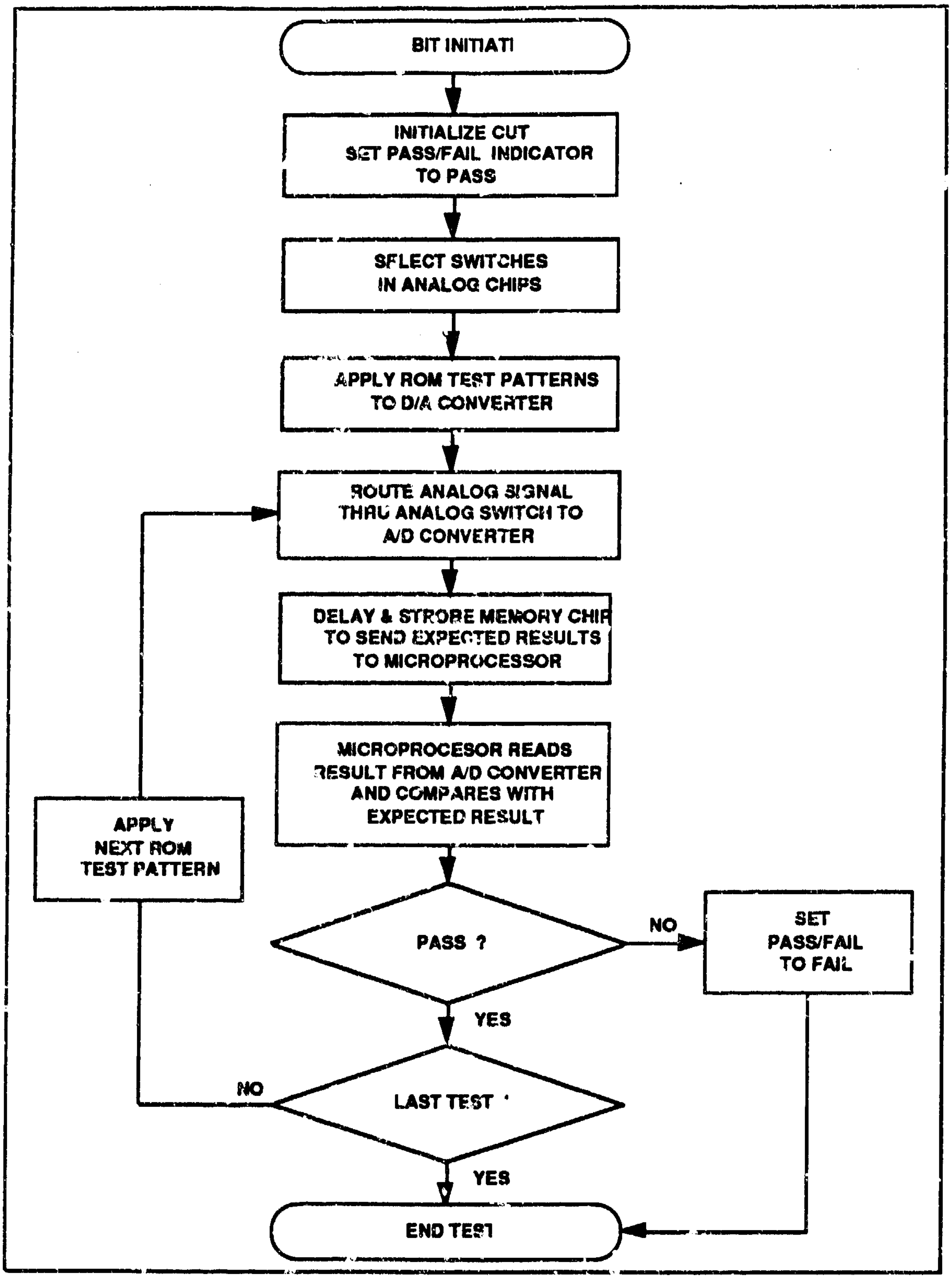

BIT INITIATI
INITIALIZE CUT SET PASS/FAIL INDICATOR TO PASS
SELECT SWITCHES IN ANALOG CHIPS
APPLY ROM TEST PATTERNS TO D/A CONVERTER
ROUTE ANALOG SIGNAL THRU ANALOG SWITCH TO A/D CONVERTER
DELAY & STRORE MEMORY CHIP TO SEND EXPECTED RESULTS TO MICROPROCESSOR
MICROPROCESOR READS RESULT FROM A/D CONVERTER AND COMPARES WITH EXPECTED RESULT
APPLY NEXT ROM TEST PATTERN
PASS ?
NO
SET PASS/FAIL TO FAIL
YES
NO
LAST TEST
YES
END TEST
274_844_104

Figure 3 BIT Sequence Flow Chart For Analog Wraparound

## BIT TECHNIQUE: ANALOG WRAPAROUND

### CATEGORY: ADVANTAGES

1. Analog wraparound only requires minimum hardware and is a conceptually simple design which is easy to implement.
2. Chips that are needed are readily available.
3. This technique may also be used in conjunction with "MICROPROCESSOR BIT", another Computer-Aided Design/Built-In-Test (CADBIT) technique, to extend the BIT coverage to include the Input/Output (I/O) chips (which are not normally checked out with the Microprocessor BIT).
4. Analog wraparound provides a rigorous test of the A/D and D/A converter interfaces, e.g.:
   a. Dynamic range
   b. Analog accuracy
   c. Conversion time
5. Test vectors can be written to duplicate realistic application commands, e.g., step response and waveform synthesis.

### CATEGORY: DISADVANTAGES

1. The technique only checks out a small portion of the LRM.
2. As the microprocessor system increases in size along with more D/A or A/D converters, then the number of analog switches must also increase (in quantity), therefore, increasing real estate and firmware requirements. Additional ROMs may be required to store the additional test patterns.

## BIT TECHNIQUE: ANALOG WRAPAROUND

## CATEGORY: ATTRIBUTES

1. CONCURRENCY

   - Test is non-concurrent with operational use.

2. TECHNOLOGY

   - Applicable to circuits using HYBRID technology, i.e. contains both digital and analog circuitry resident in the CUT.

3. CUT MICROPROCESSOR REQUIRED?

   - Yes, the CUT must include a microprocessor and ROM space for the test vectors.

4. CUT INTERNAL DESIGN REQUIRED?

   - Hardware – No.
   - Firmware – Yes, the CUT microprocessor ROM must be programmed with test vectors to implement the wraparound BIT.

5. AREA PENALTY

   - Minimal since most microprocessor LRMs have at most only a small number of D/A or A/D peripheral devices.

6. WEIGHT PENALTY

   - Increases as the number of analog channels are wrapped around.

7. POWER PENALTY

   - Increases as the number of analog channels are wrapped around.

**BIT TECHNIQUE: ANALOG WRAPAROUND**

**CATEGORY: ATTRIBUTES, Contd**

8. TIMING PENALTY

- Test Time – Increases as the number of test patterns are applied during test.
- Throughput delay – Increased to each analog channel due to the filtering effects of the analog switch and buffers added in the analog signal path.

9. RELIABILITY IMPACT

- Decreases basic reliability as the number of analog channels are wrapped around.

10. CONCEPTUAL COMPLEXITY

- Straightforward.

11. HARDWARE/SOFTWARE/FIRMWARE

- Hardware – For wraparound switch and buffers.
- Software – Not applicable. The test vectors are stored in the microprocessor ROM.
- Firmware – Microprocessor ROM is used for storage of test vectors.

12. DESIGN COST

- Hardware cost – Increases as the number of analog channels are wrapped around.
- Software – Not applicable. The test vectors are all contained in the ROM
- Firmware cost – None, if all the test vectors fit within the available space of the microprocessor.

13. MEMORY REQUIREMENTS

- Increases as the number of test vectors are added.

14. BIT CIRCUITRY SELF-TESTABLE?

- No.

15. STAND-ALONE (SELF-CONTAINED) BIT?

- Yes.

16. NOTES

- None.

## BIT TECHNIQUE: ANALOG WRAPAROUND

## CATEGORY: DEFAULT DESIGN

Refer to Figure 4 for the Default Design schematic.

The CUT consists of an 8751 microcontroller with A/D and D/A input and output interfaces. The 8751 is used in the commercial industry for a wide variety of applications, e.g. microwave ovens, childrens' toys, and automotive dashboard and engine control electronics. The 8751 contains all necessary elements for implementation of simple control systems without the addition of RAM, ROM, or I/O interface logic. 128 byte RAM and 4K EPROM is resident on the chip as well as I/O ports and interrupts.

The A/D converter (U8) is an 8 bit successive approximations type converter and accepts unipolar (0 to 5V) signals from the buffered output of the U5 operational amplifier. The A/D converter provides microprocessor compatibility with control pins for conversion execution (WR), tristate control (CS), conversion completion status (INTR), and 1 of 8 analog input channel selection (RD). The D/A conversion process is completed in 40 microseconds.

The U5 buffer amplifier incorporates a level shifting network consisting of R5, R6, R7, and a 5 volt reference to convert the bipolar (± 5 volt) output of the LRM analog input to a unipolar (0 to 5 volt) output range. A passive Zener clamping network consisting of VR1 and R8 is also provided to protect the A/D converter (U8) input from overvoltage.

The D/A converter (U7) receives parallel 8 bit digital words from port 1 of the 8751 and is latched to the D/A input register via the 8751 port 3, bit 0. The D/A converter output voltage can swing over a bipolar range (-5V to +5V). The output of the D/A converter is buffered with an operational amplifier U4.

The Analog Wraparound technique adds a wraparound analog switch (U2) and input/output buffer amplifiers (U1 and U3). When the analog switch is selected in the normal mode (U2 pin A0), the LRM input and output analog signals flow into and out of the CUT via the analog switch. In the wraparound mode, the analog switch is activated and the CUT analog output is coupled back to the CUT analog input. The LRM analog input and output lines are isolated from the CUT when the analog switch is selected in the wraparound mode. The BIT test is performed by the microprocessor which controls the analog switch condition and the application of test vectors.

The BIT INITIATE is applied to port P3 bit 2 and the PASS/FAIL results are provided at port 4 bit 4. The duration and phasing of these signals are defined in the microprocessor ROM and are at the discretion of the user.

## BIT TECHNIQUE: ANALOG WRAPAROUND

## CATEGORY: DEFAULT DESIGN, Contd

### LIMITATIONS

A noteworthy limitation of the system includes the 300 ohm nominal series "on" resistance of the analog switches. Consideration must be made to reduce loading effects in the system by usage of suitable buffer amplifiers. In this case, the AD642 operational amplifier provides insignificant loading to the analog switch output and is satisfactory for demonstration of this BIT technique.

The user should also be aware that the analog switch, when selected in the wraparound mode, deadfaces the analog input and analog outputs to the LRM. A closed loop control system will not be suitable here unless extra circuitry is added to force the LRM analog output to some fixed state while the Analog Wraparound BIT test is performed.

The input voltage and bandwidth characteristics of the analog signal are limited by the filtering effects of the AD642 buffer amplifier and analog switch.The Analog Devices Linear data book should be consulted prior to usage for any given application. No attempt is made here to recommend the specific usage of these parts as a general solution to all Analog Wraparound BIT implementations.

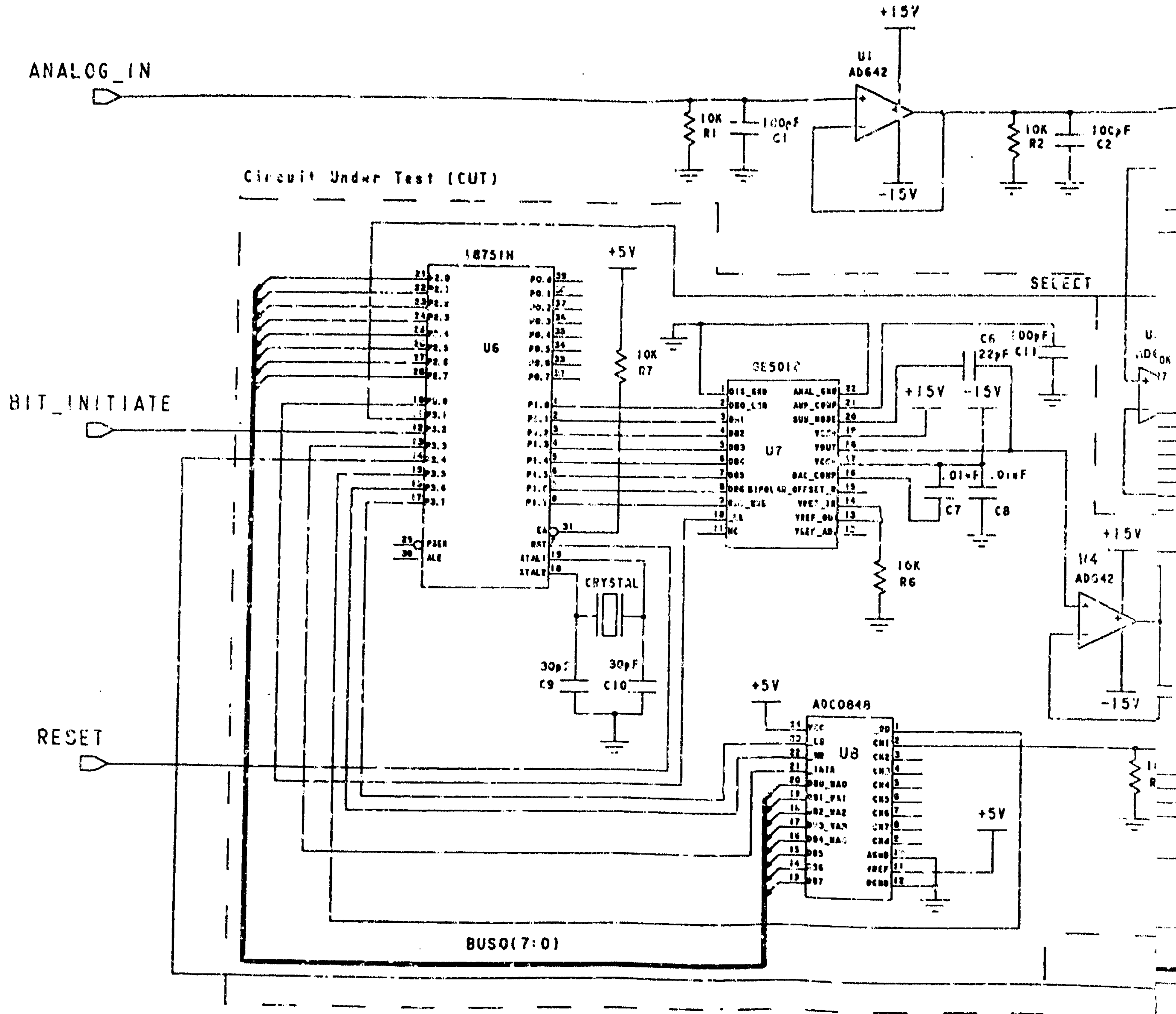

ANALOG_IN
BIT_INITIATE
RESET
Circuit Under Test (CUT)
U1
AD642
+15V
-15V
10K
R1
100pF
C1
10K
R2
100pF
C2
8751H
U6
+5V
10K
R7
SELECT
C6
22pF
100pF
C11
SE5012
U7
+15V
-15V
.01uF
.01uF
C7
C8
10K
R6
U4
AD642
+15V
-15V
CRYSTAL
30pF
C9
30pF
C10
+5V
ADC0848
U8
+5V
BUS0(7:0)

Figure 4 A

1

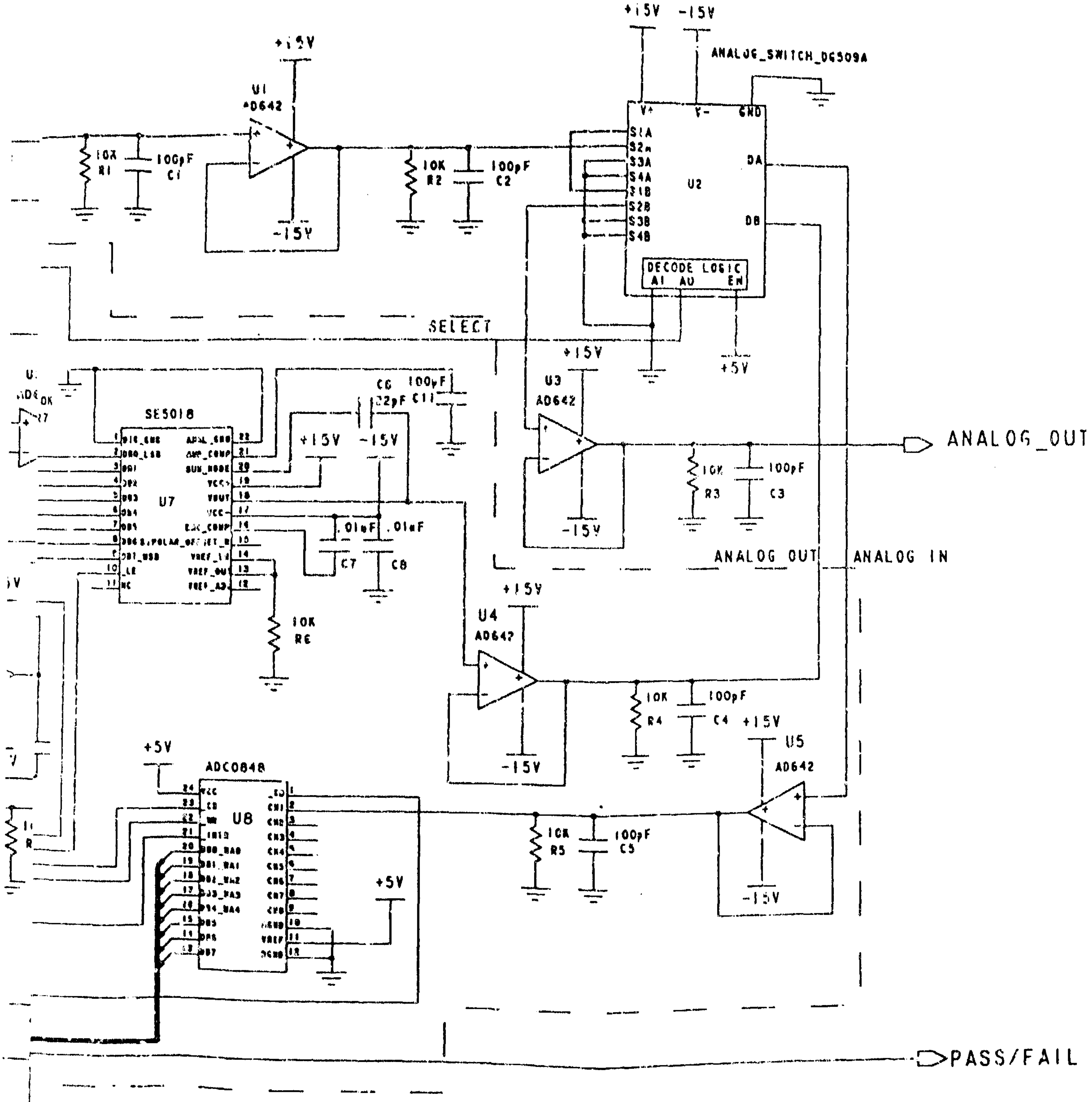


Figure 4 Analog Wraparound – Default Design

2

**BIT TECHNIQUE: ANALOG WRAPAROUND**

**CATEGORY: BIT TECHNIQUE INSERTION DIAGRAM**

Figure 5 shows the BIT Technique Insertion Diagram for this BIT technique.

BIT ELEMENT 1

Connect the CUT analog input, output, and enable to the DG509A per the default design figure 4.

Connect the LRM IN analog input and the LRM analog output to the AD642 buffers per the default design figure 4.

Connect resistors, capacitors, power, grounds, and jumpers to the AD642 buffers and DG509A per the default design figure 4.

BIT ELEMENT 2

Connect the LRM IN 'BIT INITIATE' to the CUT microprocessor input port.

BIT ELEMENT 3

Connect the LRM CUT 'PASS/FAIL' to the CUT microprocessor output port.

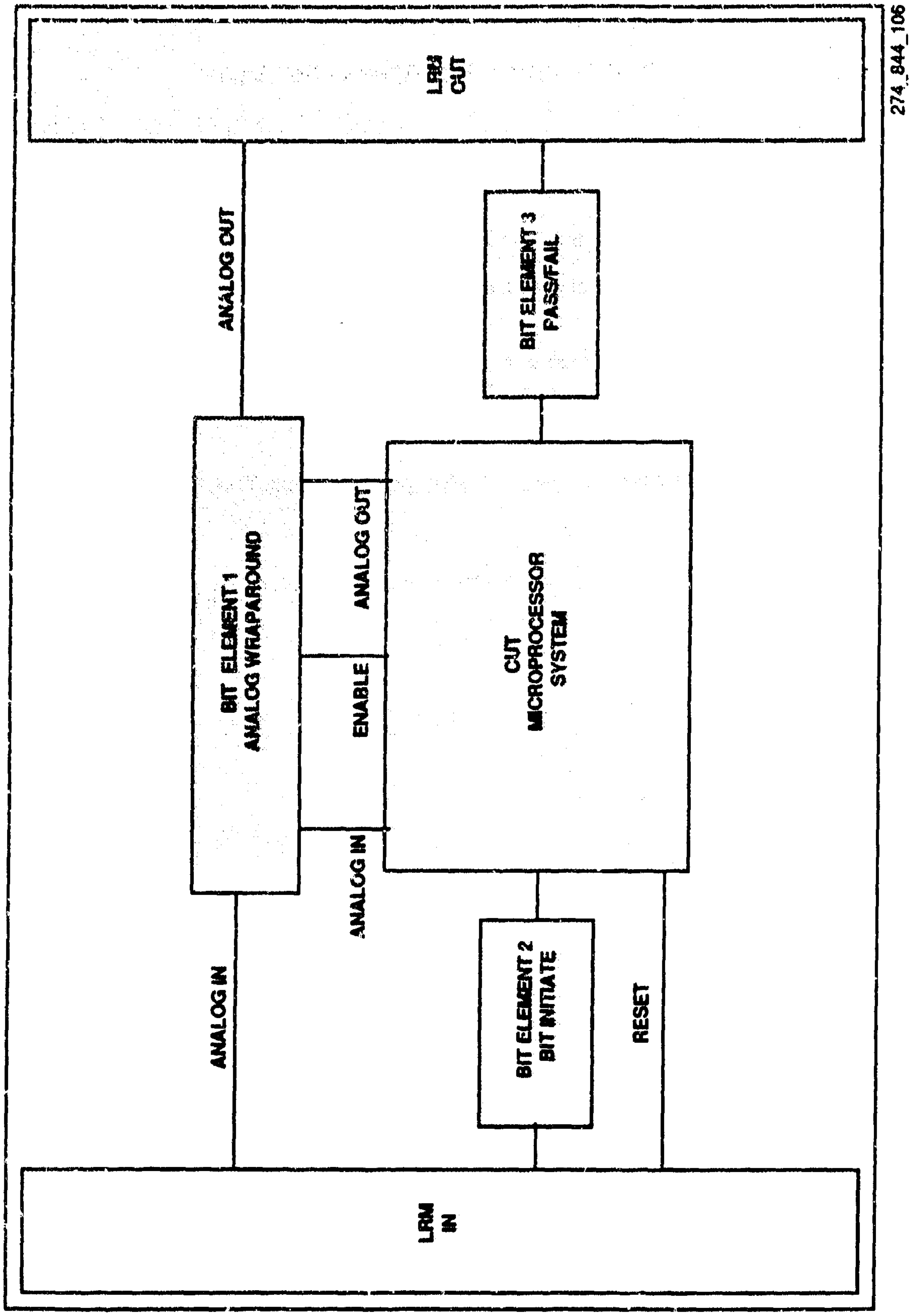


Figure 5 BIT Technique Insertion Diagram For Analog Wraparound

## BIT TECHNIQUE: ANALOG WRAPAROUND

## CATEGORY: VARIABLE DEFINITIONS

| Variable | Variable Definition | Units |
|---|---|---|
| $v_1$. | Number of CUT outputs to test | none |
| $v_2$. | #Test patterns for Analog Wraparound BIT | patterns |
| $v_3$. | Test pattern application rate for Analog Wrap BIT | patterns/sec |
| $v_4$. | CUT initialization time | sec |

## CATEGORY: COMPONENT DETERMINATION EQUATIONS

The number of units (as represented by CAD symbols) and component parts (physical packages) required to implement the Default Design are calculated as follows:

**BIT Technique Parts Table (for Default Design)**

| Index, i | Part Number | Part Type | Units Required, $u_i$ |
|---|---|---|---|
| 1 | AD642 | OP AMP (BUFFER) | $2v_1$ |
| 2 | DG509A | MUX, ANALOG | $v_1$ |
| 3 | CKR22 | CAPACITOR | $3v_1$ |
| 4 | RNR55 | RESISTOR | $3v_1$ |

The number of **Component Parts Required** is calculated (for i-th part) as follows:

$$n_i = \text{ceil}\,(u_i / upp_i)$$

**Explanation of symbols used:**

$n_i$ = Number of components (physical packages) required for i-th part
$u_i$ = Number of units (CAD symbols) required for i-th part
$upp_i$ = Number of units/package for i-th part (from Table 4.0)
ceil = Round up to nearest integer (e.g. 5.3 is rounded to 6)
$v_i$ = User-supplied value for i-th variable (see Variable Definitions)

## BIT TECHNIQUE: ANALOG WRAPAROUND

## CATEGORY: PENALTY EQUATIONS

**AREA (sq.in.)** = Sum ( $n_i * a_i$ ) + 15% for traces

**WEIGHT (gms)** = Sum ( $n_i * w_i$ ) + 10% for solder

**POWER (mW)** = Sum ( $n_i * p_i$ )

**TEST TIME (sec)** = $v_4 + ( v_2 / v_3 ) + 0.000002$

**DELAY (ns)** = 2 * tBUFFER

**Explanation of Symbols Used.**

| | | |
|---|---|---|
| Sum | = | Sum over parts in BIT Technique Parts Table (i=1 to 4) |
| $n_i$ | = | Number of component packages for i-th part in Parts Table<br>$n_i$ is calculated according to Component Determination Equations |
| $a_i$ | = | Area in sq.in. of i-th part in Parts Table |
| $p_i$ | = | Power dissipation in mW (typical) for i-th part in Parts Table |
| $w_i$ | = | Weight in grams of i-th part in Parts Table |
| $v_i$ | = | User-supplied value for i-th variable (see Variable Definitions) |
| tBUFFER | = | Max delay for OP AMP (BUFFER) from Table 4.0 |

## CATEGORY: BIBLIOGRAPHY

None required.

## 4.0 MASTER PARTS DATA

Table 4.0 is a matrix that has a list of components and its attributes. It also provides the sources where the attribute data is from. A list of the attributes and brief descriptions are shown below:

1. Component I.D.
   The number is used to identity the component.

2. Part Number
   This is the generic part number for the IC and style number for the resistor and capacitor.

3. Package Type
   The package type shows how many of the same device in a single package. It can be single, dual, octal, etc.

4. Part Description
   This is the part name with brief description of what the component is.

5. Units per Package
   This number indicates how many of the same device in a single package.

6. Number of Data Bits
   This is the data width of the component when applicable.

7. Number of Address Bits
   This is the address width of the memory or microprocessor chip.

8. Number of Pins
   This is the total number of physical pins on the component.

9. Length
   The length of the component is shown in inches.

10. Width
    The width of the component is shown in inches.

11. Area per Package
    The area of the package is calculated by multiplying the Length and Width of the component. The area is given in square inch.

12. Typical Power
    This is the typical dissipated power of the component shown in milliwatts.

13. Maximum Power
    This is the maximum dissipated power of the component shown in milliwatts.

14. Weight per Package
    This is the weight of the component package shown in grams.

15. Typical delay
    This is the typical propagation delay from the device input to output.

16. Maximum delay
    This is the maximum propagation delay from the device input to output.

17. Typical Icc
    This is the typical power supply current of the component in milliamps.

18. Maximum Icc
    This is the maximum power supply current of the component in milliamps.

19. Nominal VCC
    This is the nominal power supply voltage of the component in volts.

20. Maximum VCC
    This is the maximum allowed power supply voltage of the component in volts.

21. Reference
    The column shows the source of data extracted for this component. The page number and the title of the data book are provided.

## 4.1 PARTS/TECHNIQUE CROSS REFERENCE

Table 4.1 shows a list of components versus the BIT technique default designs using those components. To minimize the number of unique components, common technology and the same components are used across the thirteen default designs as much as possible.

## 4.2 QUESTIONS/TECHNIQUE CROSS REFERENCE

Table 4.2 is a matrix that has a list of questions versus the thirteen techniques using those questions. In the CADBIT II software, the user is expected to answer a merged set of questions corresponding to the suitable BIT techniques. The answers to the questions are captured into variables in the software and are used downstream in component determination equations and penalty equations.

## 4.3 SUITABILITY ATTRIBUTE DATA

The Suitability Selection Matrix (Figure 4.0) shows the criteria of determining suitable BIT techniques based upon the answers provided by the user to the four questions as part of the User Requested Data. The selection process is performed by elimination of unsuitable techniques. For example, in order for analog wraparound to be a suitable technique, the CUT (Circuit Under Test ) technology type is hybrid (H). In addition, the CUT allows non concurrent BIT technique, needs an available microprocessor, and doesn't care about the internal access constraint. The BIT techniques left are considered suitable techniques pending forced exclusion option by the user in the selection process.

| QUESTION | ANA WRAP | COMPARATOR | DIG WRAP | EDCC | O-B ROM | $\mu$ PROC |
|---|---|---|---|---|---|---|
| TYPE (DAH) | H | A, H | D, H | D, H | D, H | D, H |
| NON CONCURRENT OK? | Y | Y | Y | Y, N | Y | Y, N |
| CUT INTERNAL ACCESS? | Y, N | Y, N | Y, N | Y | Y, N | Y, N |
| MICROPROCESSOR AVAIL? | Y | Y, N | Y | Y, N | Y, N | |

| QUESTION | BILBO | OBIVCR | PRPG | RED | SCAN | V SUM | $\mu$ PROC |
|---|---|---|---|---|---|---|---|
| TYPE (DAH) | D, H | D, H | D, H | A, H | D, H | A, H | D, H |
| NON CONCURRENT OK? | Y | Y | Y | Y, N | Y | Y, N | Y, N |
| CUT INTERNAL ACCESS? | Y | Y | Y, N | Y, N | Y, N | Y, N | Y, N |
| MICROPROCESSOR AVAIL? | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | |

Figure 4.0 Suitability Selection Matrix

274_844_107

# TABLE 4.0 CADBIT II MASTER PARTS LIST (MPL)

| COMP ID | PART NUMBER | PKG TYPE | PART DESCRIPTION | UNITS /PKG up | #DATA BITS b | #ADDR BITS a | # OF PINS | LENGTH (in.) | WIDTH (in.) | AREA /pkg (sq.in.) | TYP PWR (mW) | MAX PWR (mW) | WEIGHT /pkg (gms) | TYP DELAY (ns) | MAX DELAY (ns) | TYP ICC (mA) | MAX ICC (mA) | NOM VCC (V) | MAX VCC (V) | REF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| U1 | 54ALS00A | QUAD | NAND, 2-INPUT | 4 | n/a | n/a | 14 | 0.785 | 0.320 | 0.251 | 5.0 | 16.5 | 2.00 | 5.3 | 16 | 1.00 | 3 | 5.0 | 5.5 | TI ALS 2-3 |
| U2 | 54ALS02 | QUAD | NOR, 2-INPUT | 4 | n/a | n/a | 14 | 0.785 | 0.320 | 0.251 | 7.8 | 22.0 | 2.00 | 8.0 | 18 | 1.51 | 4 | 5.0 | 5.5 | TI ALS 2-9 |
| U3 | 54ALS04B | HEX | INVERTER | 6 | n/a | n/a | 14 | 0.785 | 0.320 | 0.251 | 8.9 | 23.1 | 2.00 | 4.7 | 14 | 1.78 | 4.2 | 5.0 | 5.5 | TI ALS 2-15 |
| U4 | 54ALS08 | QUAD | AND, 2-INPUT | 4 | n/a | n/a | 14 | 0.785 | 0.320 | 0.251 | 8.8 | 22.0 | 2.00 | 6.0 | 18 | 1.75 | 4 | 5.0 | 5.5 | TI ALS 2-21 |
| U5 | 54ALS10A | TRIPLE | NAND, 3-INPUT | 3 | n/a | n/a | 14 | 0.785 | 0.320 | 0.251 | 3.8 | 12.1 | 2.00 | 5.3 | 16 | 0.76 | 2.2 | 5.0 | 5.5 | TI ALS 2-27 |
| U6 | 54ALS30A | SINGLE | NAND, 8-INPUT POSITIVE | 1 | 8 | n/a | 14 | 0.785 | 0.320 | 0.251 | 1.9 | 5.0 | 2.00 | 5.0 | 15 | 0.38 | 0.9 | 5.0 | 5.5 | TI ALS 2-55 |
| U7 | 54ALS32 | QUAD | OR, 2-INPUT | 4 | n/a | n/a | 14 | 0.785 | 0.320 | 0.251 | 11.3 | 27.0 | 2.00 | 8.0 | 18 | 2.25 | 4.9 | 5.0 | 5.5 | TI ALS 2-59 |
| U8 | 54ALS74A | DUAL | FLIP-FLOP, D POS-EDGE-TRIG w/PRE & CLR | 2 | n/a | n/a | 14 | 0.785 | 0.320 | 0.251 | 12.0 | 22.0 | 2.00 | 7.7 | 23 | 2.40 | 4 | 5.0 | 5.5 | TI ALS 2-77 |
| U9 | 54ALS86 | QUAD | EXCLUSIVE-OR, 2-INPUT | 4 | n/a | n/a | 14 | 0.785 | 0.320 | 0.251 | 19.5 | 32.5 | 2.00 | 7.3 | 22 | 3.90 | 5.9 | 5.0 | 5.5 | TI ALS 2-81 |
| U10 | 54ALS138 | SINGLE | DECODER/DEMUX, 3 TO 8 LINE | 1 | n/a | n/a | 16 | 0.785 | 0.320 | 0.251 | 25.0 | 55.0 | 2.00 | 9.3 | 28 | 5.00 | 10 | 5.0 | 5.5 | TI ALS 2-119 |
| U11 | 54ALS157 | QUAD | SELECTOR/MUX, 1 OF 2 LINE DATA | 1 | 4 | n/a | 16 | 0.785 | 0.320 | 0.251 | 30.0 | 60.5 | 2.00 | 10.5 | 28 | 6.00 | 11 | 5.0 | 5.5 | TI ALS 2-141 |
| U12 | 54ALS161B | SINGLE | COUNTER, SYNCHRONOUS 4-BIT BINARY | 1 | 4 | n/a | 16 | 0.785 | 0.320 | 0.251 | 80.0 | 115.5 | 2.00 | 11.3 | 34 | 12.0 | 21 | 5.0 | 5.5 | TI ALS 2-148 |
| U13 | 54ALS164 | SINGLE | SHIFT REG, 8-BIT PARALLEL OUT/SER IN | 1 | 8 | n/a | 14 | 0.785 | 0.320 | 0.251 | 50.0 | 100.0 | 2.00 | 11.0 | 33 | 10.0 | | 5.0 | 5.5 | TI ALS 2-152 |
| U14 | 54ALS165 | SINGLE | SHIFT REG, 8-BIT PARALLEL IN/SER OUT | 1 | 8 | n/a | 16 | 0.785 | 0.320 | 0.251 | 80.0 | 88.0 | 2.00 | 16.6 | 56 | 16.0 | | 5.0 | 5.5 | TI ALS 2-163 |
| U15 | 54ALS244A | OCTAL | BUFFER, w/3-STATE OUTPUTS | 2 | 4 | n/a | 20 | 0.985 | 0.320 | 0.315 | 68.3 | 148.5 | 2.50 | 9.7 | 26 | 13.67 | 27 | 5.0 | 5.5 | TI ALS 2-277 |
| U16 | 54ALS253 | DUAL | SELECTOR/MUX, 1 OF 4 LINE DATA | 2 | n/a | n/a | 16 | 0.785 | 0.320 | 0.251 | 36.0 | 77.0 | 2.00 | 10.0 | 30 | 7.00 | 14 | 5.0 | 5.5 | TI ALS 2-290 |
| U17 | 54ALS273 | OCTAL | FLIP-FLOP, D POS-EDGE-TRIGGERED W/CLR | 1 | 8 | n/a | 20 | 0.985 | 0.320 | 0.315 | 75.0 | 159.5 | 2.50 | 8.0 | 24 | 15.0 | 28 | 5.0 | 5.5 | TI ALS 2-319 |
| U18 | 54ALS299 | SINGLE | REGISTER, 8-INPUT UNIV SHIFT/STORAGE | 1 | 8 | n/a | 20 | 0.985 | 0.320 | 0.315 | 100.0 | 220.0 | 2.50 | 12.7 | 38 | 20.0 | 40 | 5.0 | 5.5 | TI ALS 2-343 |
| U19 | 54ALS373 | OCTAL | LATCH, D TRANSPARENT w/3-STATE OUTPUTS | 1 | 8 | n/a | 20 | 0.985 | 0.320 | 0.315 | 70.0 | 148.5 | 2.50 | 11.0 | 33 | 14.0 | 27 | 5.0 | 5.5 | TI ALS 2-361 |
| U20 | 54ALS374 | OCTAL | FLIP-FLOPS, D EDGE-TRIG 3-STATE OUTPUTS | 1 | 8 | n/a | 20 | 0.985 | 0.320 | 0.315 | 83.3 | 170.5 | 2.50 | 11.0 | 33 | 16.67 | 31 | 5.0 | 5.5 | TI ALS 2-369 |
| U21 | 54ALS521 | SINGLE | COMPARATOR, 8-BIT | 1 | 8 | n/a | 20 | 0.985 | 0.320 | 0.315 | 60.0 | 104.5 | 2.50 | 7.7 | 23 | 12.0 | 19 | 5.0 | 5.5 | TI ALS 2-379 |
| U22 | 54LS595 | SINGLE | SHIFT REG, 8-BIT w/OUTPUT LATCHES | 1 | 8 | n/a | 16 | 0.785 | 0.320 | 0.251 | 198.3 | 357.5 | 2.00 | [illegible] | 39 | 39.67 | 65 | 5.0 | 5.5 | TI LS 3-1153 |
| U23 | 54ALS867 | SINGLE | COUNTER, SYNC 8-BIT U/D w/ASYNC CLR | 1 | 8 | n/a | 24 | 1.290 | 0.320 | 0.410 | 142.5 | 286.0 | 3.00 | 12.4 | 37 | 28.50 | | 5.0 | 5.5 | TI ALS 2-717 |
| U24 | 27C256-12 | SINGLE | PROM, 32K x 8-BIT UV-ERASABLE | 1 | 8 | 15 | 28 | 1.450 | 0.600 | 0.870 | 7.4 | 159.0 | 3.50 | | | 15.00 | 30.0 | 5.0 | 5.5 | TI MOS 6-91 |
| U25 | 2817A | SINGLE | E²PROM, 2K x 8-BIT | 1 | 8 | 11 | 28 | 1.490 | 0.625 | 0.931 | 200.0 | 605.0 | 3.50 | | | 40.00 | 110.0 | 5.0 | 5.5 | Seeq 1-31 |
| U26 | MT5C2568 | SINGLE | RAM, 32K X 8-BIT CMOS STATIC | 1 | 8 | 15 | 28 | 1.414 | 0.310 | 0.438 | 35.0 | 550.0 | 3.50 | | | 7.00 | 100.0 | 5.0 | 5.5 | Mic4-83 |
| U27 | IDT71682 | SINGLE | RAM, 4K x 4-BIT CMOS STATIC | 1 | 4 | 12 | 24 | 1.280 | 0.320 | 0.410 | 175.0 | 605.0 | 3.00 | | | 35.00 | 110.0 | 5.0 | 5.5 | IDT 4-22 |
| U28 | 8751H | SINGLE | MICROCONTROLLER, 8-BIT WITH 4K EPROM | 1 | 8 | 16 | 40 | 2.080 | 0.605 | 1.258 | 1250.0 | 1375.0 | 12.00 | | | | 250.0 | 5.0 | 5.5 | I7-44 |
| U29 | 87C51 | SINGLE | MICROCONTROLLER, 8-BIT CHMOS w/4K EPROM | 1 | 8 | 16 | 40 | 2.080 | 0.605 | 1.258 | 58.0 | 137.5 | 12.00 | | | 11.50 | 25.0 | 5.0 | 5.5 | I7-44 |
| U30 | M55310/16-12 | SINGLE | OSCILLATOR, MICROCONTROLLER CRYSTAL | 1 | n/a | n/a | 14 | 0.887 | 0.540 | 0.479 | 100.0 | 240.0 | 14.17 | | | | 20.0 | 12.0 | 12.0 | Mfd |
| U31 | 8255A | SINGLE | PROGRAMMABLE PERIPHERAL INTERFACE | 1 | 24 | n/a | 40 | 2.060 | 0.615 | 1.279 | | 660.0 | 12.00 | | | | 120.0 | 5.0 | 5.5 | I5-273 |
| U32 | ADC0848 | SINGLE | A/D CONVERTER, 8-BIT uP W/MUX OPT | 1 | 8 | n/a | 24 | 1.290 | 0.295 | 0.381 | 5.0 | 13.8 | 7.10 | | | 1.00 | 2.5 | 5.0 | 5.5 | N3-170 |
| U33 | SE5018 | SINGLE | D/A CONVERTER, 8-BIT uP-COMPATIBLE | 1 | 8 | n/a | 22 | 1.105 | 0.399 | 0.441 | 255.0 | 435.0 | 7.10 | | | 17.00 | 29.0 | 15.0 | 15.0 | Sig4-83 |
| U34 | LM119 | DUAL | COMPARATOR, HIGH-SPEED | 2 | n/a | n/a | 14 | 0.785 | 0.320 | 0.251 | 165.0 | 240.0 | 2.00 | | | 11.00 | 16.0 | 15.0 | 15.0 | N5-34 |
| U35 | LM741 | SINGLE | OP AMP | 1 | n/a | n/a | 8 | 0.400 | 0.320 | 0.128 | | 0.0 | 1.00 | | | | | | | |
| U36 | OP-07 | SINGLE | OP AMP, ULTRALOW OFFSET VOLTAGE | 1 | n/a | n/a | 3 | 0.460 | 0.460 | 0.212 | 90.0 | 120.0 | 1.30 | | | 3.00 | 4.0 | 15.0 | 15.0 | AD 2-383 |
| U37 | AD642 | DUAL | OP AMP, PRECISION DUAL BIFET | 2 | n/a | n/a | 8 | 0.460 | 0.460 | 0.212 | 66.0 | 84.0 | 1.30 | | | 2.20 | 2.8 | 15.0 | 15.0 | AD 2-95 |
| U38 | AD524A | SINGLE | AMPLIFIER, PRECISION INSTRUMENTATION | 1 | n/a | n/a | 16 | 0.840 | 0.320 | 0.269 | 105.0 | 150.0 | 2.00 | | | 3.50 | 5.0 | 15.0 | 15.0 | AD 4-25 |
| U39 | DG509A | SINGLE | MUX, 4-CHANNEL CMOS ANALOG | 1 | 4 | n/a | 16 | 0.770 | 0.325 | 0.250 | 30.0 | 58.5 | 2.60 | | | 2.00 | 3.9 | 15.0 | 15.0 | Si 5-206 |
| U40 | HI-508 | SINGLE | MUX, 8-CHANNEL CMOS | 1 | 8 | n/a | 16 | 0.785 | 0.320 | 0.251 | 23.5 | 51.0 | 2.00 | | | 1.90 | 3.4 | 15.0 | 15.0 | H 4-16 |
| U41 | HI-5043 | DUAL | SWITCH, SPST CMOS ANALOG (note 1) | 1 | 2 | n/a | 16 | 0.785 | 0.320 | 0.251 | 1.5 | 10.5 | 2.00 | | | | 0.5 | 15.0 | 15.0 | H 3-37 |
| U42 | 8206 | SINGLE | ERROR DETECTION AND CORRECTION UNIT | 1 | 16 | n/a | 68 | 1.165 | 1.165 | 1.357 | 500.0 | 1485.0 | 12.00 | 67 | 67 | 100.0 | 270.0 | 5.0 | 5.5 | |
| U43 | RNC55H | SINGLE | RESISTOR, METAL, FILM | 1 | n/a | n/a | 2 | 0.250 | 0.109 | 0.027 | | 250.0 | 0.10 | | | | | | | |
| U44 | RNR55 | SINGLE | RESISTOR, FIXED, FILM | 1 | n/a | n/a | 2 | 0.250 | 0.109 | 0.027 | 50.0 | 125.0 | 0.10 | | | | | | | Mfd |
| U45 | RNC80 | SINGLE | RESISTOR, METAL, FILM | 1 | n/a | n/a | 2 | 0.250 | 0.109 | 0.027 | | 250.0 | 0.10 | | | | | | | |
| U46 | CKR05 | SINGLE | CAPACITOR, 30pF & .1microF | 1 | n/a | n/a | 2 | | | 0.000 | | | 0.20 | | | | | | | |
| U47 | CKR22 | SINGLE | CAPACITOR, 100pF | 1 | n/a | n/a | 2 | | | 0.000 | | | 0.20 | | | | | | | |
| U48 | CMR03 | SINGLE | CAPACITOR, FIXED, MICA DIELECTRIC | 1 | n/a | n/a | 2 | 0.270 | 0.190 | 0.051 | 0.0 | 0.0 | 0.20 | | | | | | | Mfd |

Notes

1 CADBIT symbol (="unit") for HI-5043 consists of TWO swiches (i.e. entire DUAL package)

2 TYP PWR =TYP ICC (ave) * NOM VCC (from TI databook)

3 MAX PWR =MAX ICC * MAX VCC (or 2*TYP PWR)

4 #DATA BITS =WORD LENGTH for MEMORY devices or #DATA LINES for others

5 MAX DELAY =MAX(tPLHmax,tPHLmax,tPLZmax,...)

Databooks:

TI ALS=Texas Instruments, "ALS/AS Logic Data Book", 1986

TI MOS=Texas Instruments, "MOS Memory Data Book", 1989

N=National Semiconductor "ALS/AS Logic Databook", 1987

AD=Analog Devices, "1990/91 Linear Products Databook"

Si=Siliconix,"Integrated Circuits Data Book", 1990

274_844_127

## TABLE 4.1 COMPONENT/BIT TECHNIQUE CROSS REFERENCE

| COMP ID | PART NUMBER | PKG TYPE | PART DESCRIPTION | BIT Technique Default Designs | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 1 | 2 | 3 * | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| | | | | OBR | MPB | MDIAG | OBSVCE | BILBO | EDCC | SCAN | DWRAP | PRPC | COMP | VSUM | REDUN | AWRAP |
| U1 | 54ALS00A | QUAD | NAND, 2-INPUT | | | | | x | x | | | | | | | |
| U2 | 54ALS02 | QUAD | NOR, 2-INPUT | | | | x | x | | | | | | | | |
| U3 | 54ALS04B | HEX | INVERTER | x | x | | x | x | | x | x | x | x | | | |
| U4 | 54ALS08 | QUAD | AND, 2-INPUT | x | | | x | x | | | | x | | | | |
| U5 | 54ALS10A | TRIPLE | NAND, 3-INPUT | | | | | | | | | | x | | | |
| U6 | 54ALS30A | SINGLE | NAND, 8-INPUT POSITIVE | | | | | | | | | | | | | |
| U7 | 54ALS32 | QUAD | OR, 2-INPUT | | | | | x | | x | | | | | | |
| U8 | 54ALS74A | DUAL | FLIP-FLOP, D POS-EDGE-TRIG w/PRE & CLR | x | | | x | x | x | | | x | x | x | x | |
| U9 | 54ALS86 | QUAD | EXCLUSIVE-OR, 2-INPUT | | | | x | x | | | | x | | | | |
| U10 | 54ALS138 | SINGLE | DECODER/DEMUX, 3 TO 8 LINE | | | | x | | | x | | | | | | |
| U11 | 54ALS157 | QUAD | SELECTOR/MUX, 1 OF 2 LINE DATA | x | x | | x | x | | | | x | | | | |
| U12 | 54ALS161B | SINGLE | COUNTER, SYNCHRONOUS 4-BIT BINARY | x | | | | x | | | | x | x | | | |
| U13 | 54ALS164 | SINGLE | SHIFT REG, 8-BIT PARALLEL OUT/SER IN | | | | | | | | | | | | | |
| U14 | 54ALS165 | SINGLE | SHIFT REG, 8-BIT PARALLEL IN/SER OUT | | | | | | | x | | | | | | |
| U15 | 54ALS244A | OCTAL | BUFFER, w/3-STATE OUTPUTS | x | | | | x | | x | x | x | | | | |
| U16 | 54ALS253 | DUAL | SELECTOR/MUX, 1 OF 4 LINE DATA | | | | x | | | x | | | | | | |
| U17 | 54ALS273 | OCTAL | FLIP-FLOP, D POS-EDGE-TRIGGERED W/CLR | | | | | | | | | | | | | |
| U18 | 54ALS299 | SINGLE | REGISTER, 8-INPUT UNIV SHIFT/STORAGE | | | | | | | | | | | | | |
| U19 | 54ALS373 | OCTAL | LATCH, D TRANSPARENT w/3-STATE OUTPUTS | | | | x | | | x | | | | | | |
| U20 | 54ALS374 | OCTAL | FLIP-FLOPS, D EDGE-TRIG 3-STATE OUTPUTS | | | | | | | | | | | | | |
| U21 | 54ALS521 | SINGLE | COMPARATOR, 8-BIT | x | | | | x | | | | x | | | | |
| U22 | 54LS595 | SINGLE | SHIFT REG, 8-BIT w/OUTPUT LATCHES | | | | | | | x | | | | | | |
| U23 | 54ALS867 | SINGLE | COUNTER, SYNC 8-BIT U/D W/ASYNC CLR | | | | | | | | | | | | | |
| U24 | 27C256-12 | SINGLE | PROM, 32K x 8-BIT UV-ERASABLE | x | x | | | | | | | | | | | |
| U25 | 2817A | SINGLE | EEPROM, 2K x 8-BIT | | | | x | | | x | | | | | | |
| U26 | MT5C2568 | SINGLE | RAM, 32K X 8-BIT CMOS STATIC | | | | | | | | | | | | | |
| U27 | IDT71682 | SINGLE | RAM, 4K x 4-BIT CMOS STATIC | | | | | | x | | | | | | | |
| U28 | 8751H | SINGLE | MICROCONTROLLER, 8-BIT WITH 4K EPROM | | | | x | | | | | | | | | |
| U29 | 87C51 | SINGLE | MICROCONTROLLER, 8-BIT CHMOS w/4K EPROM | | | | | | | x | | | | | | |
| U30 | M55310/18-12 | SINGLE | OSCILLATOR, MICROCONTROLLER CRYSTAL | | | | x | | | x | | | | | | |
| U31 | C255A | SINGLE | PROGRAMMABLE PERIPHERAL INTERFACE | | x | | | | | | | | | | | |
| U32 | ADC0846 | SINGLE | A/D CONVERTER, 8-BIT uP W/MUX OPT | | | | | | | | | | | | | |
| U33 | SE5018 | SINGLE | D/A CONVERTER, 8-BIT uP-COMPATIBLE | | | | | | | | | | | | | |
| U34 | LM119 | DUAL | COMPARATOR, HIGH-SPEED | | | | | | | | | | x | x | x | |
| U35 | LM741 | SINGLE | OP AMP | | | | | | | | | | | | | |
| U36 | OP-07 | SINGLE | OP AMP, ULTRALOW OFFSET VOLTAGE | | | | | | | | | | | x | | |
| U37 | AD642 | DUAL | OP AMP, PRECISION DUAL BIFET | | | | | | | | | | | | | x |
| U38 | AD524A | SINGLE | AMPLIFIER, PRECISION INSTRUMENTATION | | | | | | | | | | | | x | |
| U39 | DG509A | SINGLE | MUX, 4-CHANNEL CMOS ANALOG | | | | | | | | | | | | | x |
| U40 | HI 508 | SINGLE | MUX, 8-CHANNEL CMOS | | | | | | | | | | x | | | |
| U41 | HI-5043 | DUAL | SWITCH, SPST CMOS ANALOG (note 1) | | | | | | | | | | x | | | |
| U42 | I8206 | SINGLE | ERROR DETECTION AND CORRECTION UNIT | | | | | | x | | | | | | | |
| U43 | RNC55H | SINGLE | RESISTOR, METAL, FILM | | | | | | x | x | | | x | x | | |
| U44 | RNR55 | SINGLE | RESISTOR, FIXED, FILM | x | | | x | | | | | | | | x | x |
| U45 | RNC60 | SINGLE | RESISTOR, METAL, FILM | | | | | | | | | | | x | | |
| U46 | CKR05 | SINGLE | CAPACITOR, FIXED, CERAMIC DIELEC 30&100pF | | | | | | | x | | | | | | |
| U47 | CKR22 | SINGLE | CAPACITOR, FIXED, CERAMIC DIELEC 100pF | | | | | | | | | | | | | x |
| U48 | CMR03 | SINGLE | CAPACITOR, FIXED, MICA DIELECTRIC | | | | x | | | | | | | | | |

* DEFAULT DESIGN NOT REQUIRED FOR MICRODIAGNOSTICS BIT TECHNIQUE

274_844_128

TABLE 4.2 MASTER QUESTION LIST

| | QUESTIONS | BIT TECHNIQUES | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| | | OBR | MPB | MDIAG | OBIVCB | BILBO | EDCC | SCAN | DWRAP | PRPG | COMP | VSUM | REDUN | AWRAP |
| 1 | CUT initialization time (secs) ? | X | | | | X | | | X | X | | | | X |
| 2 | Number of CUT memory locations to test ? | | | | | | X | | | | | | | |
| 3 | Number of peripheral functions to test ? | | X | | | | | | | | | | | |
| 4 | Microprocessor execution speed (ips) ? | | X | | | | | | | | | | | |
| 5 | Number of CUT inputs to test ? | X | | | | X | | X | | X | X | | | |
| 6 | Number of CUT outputs to test ? | X | | | | X | | X | X | X | X | X | X | X |
| 7 | Test time per monitored voltage (ns) ? | | | | | | | | | | X | | | |
| 8 | Max propagation delay through CUT (ns) ? | X | | | | X | | | | X | | | | |
| 9 | Number of test patterns for Scan BIT ? | | | | | | | X | | | | | | |
| 10 | Clock rate for Scan BIT (kb/s) ? | | | | | | | X | | | | | | |
| 11 | Board area of Standard for Redundancy BIT (sq.in.) ? | | | | | | | | | | | | X | |
| 12 | Power (typical) of Standard for Redundancy BIT (mW) ? | | | | | | | | | | | | X | |
| 13 | Weight of Standard for Redundancy BIT (gms) ? | | | | | | | | | | | | X | |
| 14 | Additional ROMs required for Microdiagnostics BIT (#ROMs) ? | | | X | | | | | | | | | | |
| 15 | Microdiagnostics ROM typical power consumption (mW) ? | | | X | | | | | | | | | | |
| 16 | Microdiagnostics ROM area (sq.in.) ? | | | X | | | | | | | | | | |
| 17 | Microdiagnostics ROM weight (gms) ? | | | X | | | | | | | | | | |
| 18 | Microdiagnostics BIT execution time (ns) ? | | | X | | | | | | | | | | |
| 19 | Number of test patterns for Analog Wraparound BIT ? | | | | | | | | | | | | | X |
| 20 | Test pattern application rate for Analog Wraparound BIT (patterns/sec) ? | | | | | | | | | | | | | X |
| 21 | Number of test patterns for Digital Wraparound BIT (bytes) ? | | | | | | | | X | | | | | |
| 22 | Test pattern application rate for Digital Wraparound BIT (bytes/sec) ? | | | | | | | | X | | | | | |
| 23 | Number of test patterns for On Board Rom BIT ? | X | | | | | | | | | | | | |
| 24 | Memory data width (bits) ? | | | | | | X | | | | | | | |
| 25 | Number of bytes for Microprocessor BIT software in additional ROM (bytes) ? | | X | | | | | | | | | | | |
| 26 | Number of dynamic instructions for Microprocessor BIT software ? | | X | | | | | | | | | | | |
| 27 | Data bus width for Microprocessor BIT (bytes) ? | | X | | | | | | | | | | | |

274_844_129

## 5.0 MASTER SYMBOL LIST

Table 5.0 is a matrix that has a list of component symbols versus the corresponding component library where they locate. In the CADBIT II software, the mil_als_lib was selected for digital components mainly because of its availability at the early release of Mentor Graphics 8.0. The Mentor spice_lib contains basic component types like resistor, capacitor etc. For more sophisticated analog components, Mentor accuparts_lib is a good source. Logic Automation Incorporated, an open door partner of Mentor Graphics provides SmartModel Library which contains specialized models for complex ICs such as microcontrollers and interface devices. When a device does not exist in any of the available libraries, Mentor Graphics has application software allowing user to create symbol and build the corresponding model file to meet the device performance.

# TABLE 5.0 MASTER SYMBOL LIST

| COMP ID | PART NUMBER | PKG TYPE | PART DESCRIPTION | MENTOR mil_als_lib | MENTOR mil_ls_lib | MENTOR accparts_lib | MENTOR mm_lib | MENTOR spice_lib | LAI SmartModel Lib | created symbol |
|---|---|---|---|---|---|---|---|---|---|---|
| U1 | 54ALS00A | QUAD | NAND, 2-INPUT | X | | | | | | |
| U2 | 54ALS02 | QUAD | NOR, 2-INPUT | X | | | | | | |
| U3 | 54ALS04B | HEX | INVERTER | X | | | | | | |
| U4 | 54ALS08 | QUAD | AND, 2-INPUT | X | | | | | | |
| U5 | 54ALS10A | TRIPLE | NAND, 3-INPUT | X | | | | | | |
| U6 | 54ALS30A | SINGLE | NAND, 8-INPUT POSITIVE | X | | | | | | |
| U7 | 54ALS32 | QUAD | OR, 2-INPUT | X | | | | | | |
| U8 | 54ALS74A | DUAL | FLIP-FLOP, D POS-EDGE-TRIG w/PRE & CLR | X | | | | | | |
| U9 | 54ALS86 | QUAD | EXCLUSIVE-OR, 2-INPUT | X | | | | | | |
| U10 | 54ALS138 | SINGLE | DECODER/DEMUX, 3 TO 8 LINE | X | | | | | | |
| U11 | 54ALS157 | QUAD | SELECTOR/MUX, 1 OF 2 LINE DATA | X | | | | | | |
| U12 | 54ALS161B | SINGLE | COUNTER, SYNCHRONOUS 4-BIT BINARY | X | | | | | | |
| U13 | 54ALS164 | SINGLE | SHIFT REG, 8-BIT PARALLEL OUT/SER IN | X | | | | | | |
| U14 | 54ALS165 | SINGLE | SHIFT REG, 8-BIT PARALLEL IN/SER OUT | X | | | | | | |
| U15 | 54ALS244A | OCTAL | BUFFER, w/3-STATE OUTPUTS | X | | | | | | |
| U16 | 54ALS253 | DUAL | SELECTOR/MUX, 1 OF 4 LINE DATA | X | | | | | | |
| U17 | 54ALS273 | OCTAL | FLIP-FLOP, D POS-EDGE-TRIGGERED W/CLR | X | | | | | | |
| U18 | 54ALS299 | SINGLE | REGISTER, 8-INPUT UNIV SHIFT/STORAGE | X | | | | | | |
| U19 | 54ALS373 | OCTAL | LATCH, D TRANSPARENT w/3-STATE OUTPUTS | X | | | | | | |
| U20 | 54ALS374 | OCTAL | FLIP-FLOPS, D EDGE-TRIG 3-STATE OUTPUTS | X | | | | | | |
| U21 | 54ALS521 | SINGLE | COMPARATOR, 8-BIT | X | | | | | | |
| U22 | 54LS595 | SINGLE | SHIFT REG, 8-BIT w/OUTPUT LATCHES | | X | | | | | |
| U23 | 54ALS867 | SINGLE | COUNTER, SYNC 8-BIT U/D W/ASYNC CLR | X | | | | | | |
| U24 | 27C256-12 | SINGLE | PROM, 32K x 8-BIT UV-ERASABLE | | | | X | | | |
| U25 | 2817A | SINGLE | EEPROM, 2K x 8-BIT | | | | | | | X |
| U26 | MT5C2568 | SINGLE | RAM, 32K X 8-BIT CMOS STATIC | | | | | | | X |
| U27 | IDT71682 | SINGLE | RAM, 4K x 4-BIT CMOS STATIC | | | | | | X | |
| U28 | I8751H | SINGLE | MICROCONTROLLER, 8-BIT WITH 4K EPROM | | | | | | X | |
| U29 | I87C51 | SINGLE | MICROCONTROLLER, 8-BIT CHMOS w/4K EPROM | | | | | | X | |
| U30 | M55310/18-12 | SINGLE | OSCILLATOR, MICROCONTROLLER CRYSTAL | | | | | | | X |
| U31 | 8255A | SINGLE | PROGRAMMABLE PERIPHERAL INTERFACE | | | | | | X | |
| U32 | ADC0818 | SINGLE | A/D CONVERTER, 8-BIT uP W/MUX OPT | | | | | | | X |
| U33 | SE5018 | SINGLE | D/A CONVERTER, 8-BIT uP-COMPATIBLE | | | | | | | X |
| U34 | LM119 | DUAL | COMPARATOR, HIGH-SPEED | | | X | | | | |
| U35 | LM741 | SINGLE | OP AMP | | | X | | | | |
| U36 | OP-07 | SINGLE | OP AMP, ULTRALOW OFFSET VOLTAGE | | | X | | | | |
| U37 | AD642 | DUAL | OP AMP, PRECISION DUAL BIFET | | | X | | | | |
| U38 | AD524A | SINGLE | AMPLIFIER, PRECISION INSTRUMENTATION | | | X | | | | |
| U39 | DG509A | SINGLE | MUX, 4-CHANNEL CMOS ANALOG | | | X | | | | |
| U40 | HI-508 | SINGLE | MUX, 8 CHANNEL CMOS | | | | | | | X |
| U41 | HI-5043 | DUAL | SWITCH, SPST CMOS ANALOG (note 1) | | | | | | | X |
| U42 | I82C5 | SINGLE | ERROR DETECTION AND CORRECTION UNIT | | | | | | | X |
| U43 | RNC55H | SINGLE | RESISTOR, METAL, FILM | | | | | X | | |
| U44 | RNR55 | SINGLE | RESISTOR, FIXED, FILM | | | | | X | | |
| U45 | RNC60 | SINGLE | RESISTOR, METAL, FILM | | | | | X | | |
| U46 | CKR05 | SINGLE | CAPACITOR, FIXED, CERAMIC DIELEC 30&100pF | | | | | X | | |
| U47 | CKR22 | SINGLE | CAPACITOR, FIXED, CERAMIC DIELEC 100pF | | | | | X | | |
| U48 | CMR03 | SINGLE | CAPACITOR, FIXED, MICA DIELECTRIC | | | | | X | | |

274_844_130

## 6.0 BIT TECHNIQUE LITERATURE RESEARCH

A literature research was conducted utilizing the resources of the GD corporate libraries and two major California State Universities. This literature search concentrated on testability and BIT techniques described in the literature from 1986 to 1991. This restricted search was possible since the previous CADBIT program had performed a literature search prior to this period.

Over 250 articles and documents dealing with the latest BIT were identified from the abstracts. From these articles, 85 were selected, collected, reprinted, and distributed to GD test engineers. These engineers reviewed the data for two basic purposes. First, to identify additional BIT techniques that could be included in the BIT library; and second, to provide updated information for any of the 13 baselined techniques.

The breakdown of the results of the literature search by different BIT technique type is shown in Figure 6.0. The majority of the most recent articles dealt with some form of boundary scan and ASICs. Some of the latest trends are described below:

PARTIAL SCAN--Partial scan is a variation of full scan technology. In practice, users can either start with full scan and then remove the more intrusive scan elements to reduce penalty or launch a new design with less than one hundred percent scan. Growing numbers of design engineers favor partial scan over full scan technology because it offers lower area overhead and less impact on critical timing paths. With partial scan, carefully selected sequential elements in an IC design are made controllable and observable by replacing them with scannable sequential elements. ASIC and IC designers never like the area, power, and speed overhead that comes with adding flip-flops to every serial scan chain in a circuit. An ASIC design where a 5% increase in area would require the design to move to another package size and a high speed design that can afford very limited timing penalty in its critical path are good candidates for partial scan. As a result, partial scan technology is preferred by design application tightly constrained by area and performance requirements.

BOUNDARY SCAN—Boundary scan is still the best tool for improving the testability of printed circuit boards. In practice, virtually all printed circuit boards designed with boundary scan devices also contain conventional, non-scan parts. As long as this mix persists, test engineers will want to retain some in circuit test points even when boundary scan testing is used. The tough task, especially for complex high-pin-count boards, is deciding which test points should be kept and which can be dropped without sacrificing fault coverage. In this respect, boundary scan is still evolving.

QUIESCENT CURRENT TEST—I (ddq) testing takes advantage of the CMOS property that normal standby current is miniscule but escalates by orders of magnitude when defects are present. The ideal circuit for I (ddq) testing is fully static and fully complementary. This test will lead to false defect detection for fast dynamic logic, buses with undriven states, pull ups and pull-downs, and pre-change circuitry. I (ddq) testing takes into account the physical defect behavior in CMOS which other traditional stuck-at fault models lack. It appears that it may start to replace large portions of functional and scan-test vector sequences.

ANALOG SCAN—Since analog signals are continuous, the scan approach is different from the digital scan. One current approach is to insert test pins, control logic, and multiplexer cells into the mixed signal ASIC during the design stage to make the test result observable. Analog scan design makes the hidden internal circuitry easily accessible for test. Currently, major effort in testing and software development is trying to develop a similar standard to IEEE1149.1 digital scan test standard.

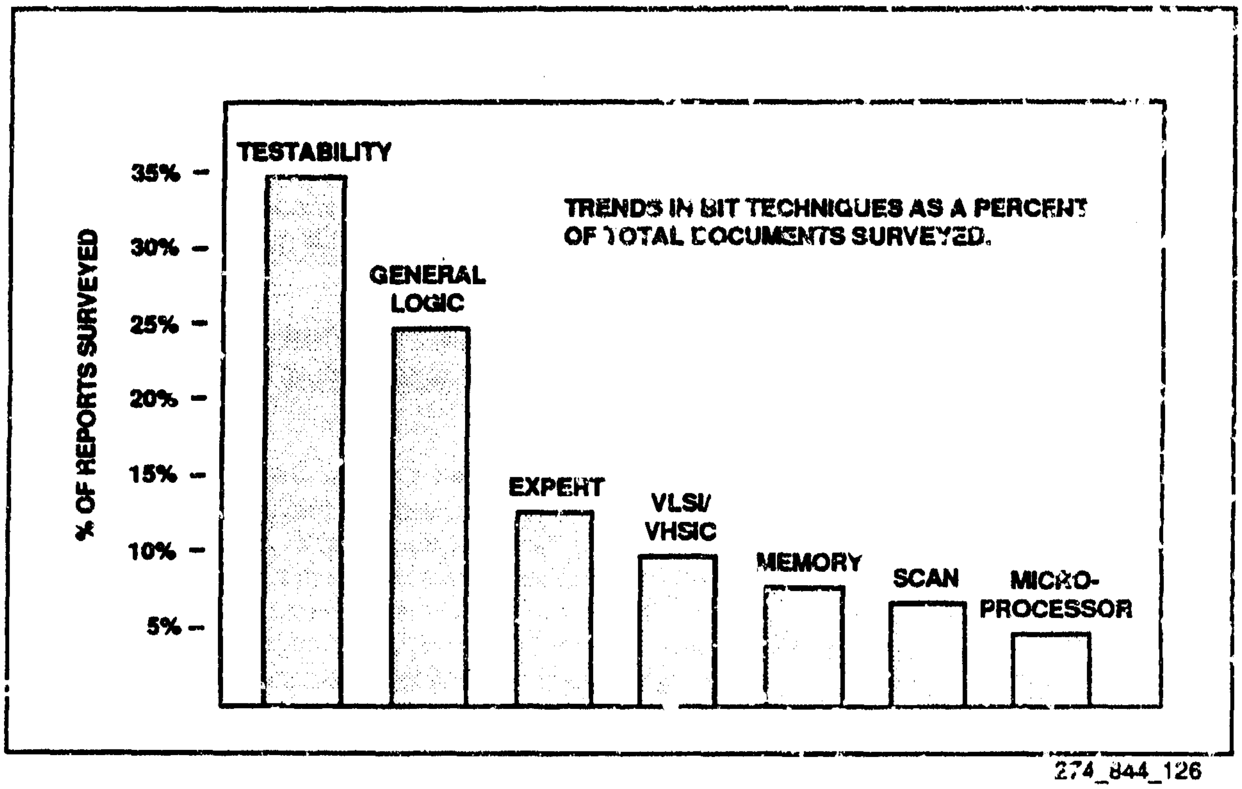


Figure 6.0 Literature Research Results And Trends

All data from the literature search were collected, reprinted, and stored in the CADBIT II technical data file. Additional documents, mainly from recent magazine articles, were obtained and reviewed throughout the program development. Relevant articles were added to the technical data files and the CADBIT II bibliography. At the end of the CADBIT II technical effort, more than 85 articles have been filed.

The CADBIT II bibliography was compiled and distributed at the First Technical Interface Meeting (TIM) and has been updated throughout the program development.

# MISSION

# OF

# ROME LABORATORY

*Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence ($C^3I$) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of $C^3I$ systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.*